Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCA

Juraj Šimlovič

Automatická kontrola prekladu

Ústav formální a aplikované lingvistiky

Vedúci diplomovej práce: RNDr. Vladislav Kuboň, Ph.D.

Študijný program: Informatika, softwarové systémy

Rád by som poďakoval vedúcemu práce, RNDr. Vladislavovi Kuboňovi, Ph.D, za výber témy, cenné rady pri písaní, názory na výsledky i trpezlivosť. Ďalej by som rád poďakoval pánovi Štěpánovi Kvapilíkovi z firmy Hieronymus za poskytnutie cenných prekladových pamätí, vďaka ktorým som mohol výsledky svojej práce reálne overiť. V neposlednej rade by som rád poďakoval i všetkým, ktorý ma pri práci podporovali a vychádzali mi v ústrety.

Prehlasujem, že som svoju diplomovú prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím s požičiavaním práce.

V Prahe dňa 6. decembra 2010 Juraj Šimlovič

Obsah

Abstrakt

Názov práce: *Automatická kontrola prekladu*
Autor: *Juraj Šimlovič*

Katedra (ústav): *Ústav formální a aplikované lingvistiky*
Vedúci diplomovej práce: *RNDr. Vladislav Kuboň, Ph.D.*
E-mail vedúceho: *Vladislav.Kubon@mff.cuni.cz*

Abstrakt: *Prekladové pamäte sa stávajú u profesionálnych prekladateľov čím ďalej, tým viac populárne; a to predovšetkým v oblasti lokalizácie produktov a pri preklade odborných či oficiálnych dokumentov. Aj keď komerčné systémy pracujúce s prekladovými pamäťami ponúkajú limitované možnosti automatickej kontroly prekladu, obvykle sa jedná iba o jednoduché nástroje vyhľadávania v texte. Navyše žiaden z týchto systémov neponúka adekvátne možnosti zapojiť do kontroly morfológiu. Profesionálny prekladatelia by ocenili automatizovaný nástroj, ktorý by ponúkal možnosti kontroly prekladu na základe pokročilejších pravidiel, a ktorý by vzal v úvahu českú, ale i anglickú morfológiu. Užitočná by bola nielen kontrola použitia správnej terminológie, ale taktiež kontrola použitia zakázaných kombinácií slov. Táto práca skúma typy chýb, ktoré prekladatelia zvyknú robiť a ponúka prehľad existujúcich riešení automatickej kontroly prekladu pre iné jazyky. Následne navrhuje a implementuje aplikáciu, ktorá sa pokúša hľadať niektoré z najčastejších chýb pri preklade do češtiny, vnášajúc morfológiu do vyhľadávacieho procesu.*

Kľúčové slová: *kontrola prekladu, automatická kontrola, prekladová pamäť*

Abstract

Title: *Automatic Checking of Translation*
Author: *Juraj Šimlovič*

Department: *Institute of Formal and Applied Linguistics*
Supervisor: *RNDr. Vladislav Kuboň, Ph.D.*
Supervisor's e-mail address: *Vladislav.Kubon@mff.cuni.cz*

Abstract: *Translation memories are becoming more and more popular with professional translators nowadays, especially in fields of software localization and translation of technical and official documents. Although commercial systems, which employ memory translation, provide some limited capabilities for automatic checking of translations, these are mostly of simple search-and-replace type. And none of these systems provide reasonable means of applying Czech morphology while checking. Professional translators could benefit from an automatic tool, which would provide more advanced rule-based checking capabilities, taking Czech and even English morphology into the process. Checking not only for correct use of terminology, but also for illicit translations and use of forbidden terms would be useful. This thesis investigates types of mistakes translators tend to make. Review of existing solutions for automatic translation checking for different languages is provided. An application is then suggested and developed, which attempts to search for some of the most frequent mistakes made in translations into Czech language, taking morphology into account while searching.*

Keywords: *translation checking, automatic checking, translation memory*

1 Úvod

Prekladatelia, podobne ako iný ľudia, ktorí vytvárajú množstvá textu, majú vo zvyku kontrolovať po sebe pred odovzdaním vytvorený text, často i niekoľkonásobne, v snahe nájsť rôzne druhy chýb a preklepov, ktoré pri preklade môžu vzniknúť, či už z nepozornosti, únavy alebo nedostatku skúseností. Táto snaha o vytvorenie bezchybného prekladu je však limitovaná množstvom času, ktorý je prekladateľ kontrole ochotný obetovať, a často i výškou odmeny, ktorú prekladateľ za preklad dostane. Prekladatelia preto radi vítajú akúkoľvek pomoc, ktorá im ponúka zvýšenie kvality prekladu, prípadne ušetrí čas strávený takouto kontrolou.

V dnešnej dobe, kedy počítače zohrávajú pri preklade stále väčšiu úlohu, je prirodzené, že prekladatelia vyhľadávajú automatizované nástroje, ktoré by ich dokázali na chyby upozorňovať. I keď takéto nástroje dnes ešte nedokážu zaručiť dokonalý výsledok, i čiastočný úspech je považovaný za prínos – pre prekladateľa to znamená, že vo výsledku zostane menej chýb.

Dostupné nástroje, ktoré dokážu prekladateľovi pri kontrole pomôcť, však obvykle ponúkajú buď iba jednoduché vyhľadávacie možnosti a nerozumejú samotnému textu, gramatike ani morfológií jazyka; alebo naopak nie sú bilingválne, takže nekontrolujú správnosť prekladu, ale iba gramatickú či pravopisnú korektnosť výsledných viet.

Cieľom tejto práce je preskúmať, aké možnosti pri preklade do češtiny dnes už existujú, aké majú výhody a aké majú nedostatky, a pokúsiť sa navrhnúť a vytvoriť nástroj, ktorý by aspoň časť týchto nedostatkov riešil.

2 Automatický preklad použitím prekladových pamätí

Cieľom automatického prekladu, a taktiež automatického prekladu použitím prekladových pamätí, je predovšetkým urýchlenie práce prekladateľa a zvýšenie efektivity jeho práce. Základná myšlienka použitia prekladových pamätí je v tom, že prekladateľ sa s rovnakým typom viet stretá počas svojej práce opakovane. Rovnaké, prípadne podobné vety pritom prekladateľ zvykne prekladať rovnakým, resp. podobným spôsobom.

Toto platí napríklad (nielen, ale predovšetkým) pri opakovanej lokalizácií vzájomne nasledujúcich verzií počítačových produktov a ich dokumentácie. Tu sa totiž medzi jednotlivými verziami väčšina textu v zásade nemení. Produkty bývajú dlhodobo zamerané na konkrétne skupiny funkcií a dopredu definované spôsoby použitia. Jednotlivé funkcie produktu sa v priebehu jeho existencie obvykle rozširujú, a nie naopak. Lokalizácia a hlavne dokumentácia takýchto produktov je tak viac-menej rozširovaná o nové prvky, pričom popis starých prvkov ostáva nezmenený, a to často i po dobu niekoľkých verzií až celej dĺžky existencie produktu.

Automatický preklad použitím prekladových pamätí efektívne zacieľuje práve túto podobnosť medzi starými a novými verziami dokumentov. Jej primárnym cieľom je to, aby vety a názvy, ktoré lokalizačný tým už raz preložil, nebolo nutné prekladať v novej verzií opakovane. Pokiaľ zvládnu prekladové pamäte správne preložiť to, čo sa medzi jednotlivými verziami nezmenilo, práca na lokalizácií novej verzie sa efektívne zmenší na lokalizovanie zmien medzi danými verziami.

2.1 Prekladová pamäť

Prekladová pamäť všeobecne je súbor dvojíc viet (resp. častí viet), kde prvá z dvojice je v jazyku primárnom a druhá v jazyku sekundárnom.[1] I keď smer prekladu nemusí byť prekladovou pamäťou priamo vynútený, je obvyklé, že primárny jazyk je jazyk originálneho dokumentu, z ktorého sa prekladá; a sekundárny jazyk je jazyk cieľového dokumentu do ktorého sa prekladá. Príkladom dvojice z prekladovej

[1] Používa sa tiež pomenovanie „zdrojový jazyk“ a „cieľový jazyk“, anglicky „source language“ a „target language“.

pamäte je napríklad dvojica („Create new document“, „Vytvor nový dokument“), alebo tiež dvojica („&File“, „&Súbor“).[1]

Takáto dvojica tvorí v prekladovej pamäti jeden záznam. Záznamov môže byť v jednej pamäti stovky[2], obvykle státisíce, a v prípade dlhodobo zbieraných prekladových pamätí veľkých spoločností i milióny. Medzi jednotlivými záznamami z prekladovej pamäte nie je obvykle udržovaný žiaden priamy súvis.

Pri preklade dokumentu použitím prekladových pamätí používa prekladateľ program, ktorý vety (resp. názvy, resp. časti viet) z originálu vyhľadáva v záznamoch z prekladovej pamäte.[3] Pokiaľ program nájde zhodu s nejakým záznamom v prekladovej pamäti, predloží tento prekladateľovi ako návrh na možný preklad. Prekladateľ návrhy následne prijíma, alebo upravuje v prípade, že navrhovaný preklad nie je úplne vyhovujúci.

Novo preložené vety (v prípade, že prekladateľ návrh na preklad miesto prijatia významne upraví, alebo dokonca kompletne prepíše) sú obvykle do prekladovej pamäte vkladané ako nové záznamy. Prekladová pamäť sa takto automaticky rozširuje o nové informácie a poskytuje v budúcnosti viac efektívnejších návrhov na preklad.

---

[1] Zakiaľ v prvom príklade je dvojicou obyčajná veta „Create new document“, ktorá môže byť súčasťou dokumentácie alebo tzv. tooltip nápovedí; v druhom príklade nesie dvojica v sebe zároveň i funkčnú informáciu o skratkovej klávese, ktorú bude užívateľ môcť použiť, aby sa k funkcií „File“, resp. „Súbor“ dostal použitím klávesnice. V operačnom systéme Microsoft Windows je totiž v dialógoch a v aplikačnom menu dovolené používať znak „&“ na označenie tzv. „Mnemonics”, čiže skratkových písmen. V tomto prípade zápis „&File“ znamená, že položka menu bude obsahovať v anglickej verzií text „File“; písmeno „F“ bude podčiarknuté; a kombinácia kláves „Alt+F“ bude automaticky fungovať ako mnemonics, takže skratkový kláves. V prípade prekladu „&Súbor“ to ale znamená, že položka menu bude obsahovať v preloženej verzií text „Súbor“; podčiarknuté bude písmeno „S“; a ako skratkový kláves bude automaticky fungovať kombinácia „Alt+S“. Na tomto príklade je ilustrované, že lokalizácia produktov pre konkrétny systém môže vďaka základnej funkcií daného systému zahrňovať i lokalizáciu funkčnosti produktu (v tomto prípade lokalizáciu skratkových kláves).

[2] Pokiaľ sa jedná o prekladovú pamäť pre konkrétny produkt. Takto malé pamäte však nebývajú zvykom a jedná sa obvykle o testovacie verzie prekladu, skúšobné projekty spoločností a počiatočné fáze vývoja produktov. (Flournoy, et al., 2009)

[3] Príkladom programov, ktoré s prekladovými pamäťami pracujú, sú napríklad komerčné produkty SDLX (formát XLIFF), TRADOS and Déjà Vu (formát TMX).

2.2 Existujúce produkty a ich obvyklé pokročilé funkcie

Rôzne typy existujúcich produktov, umožňujúcich preklad pomocou prekladových pamätí, poskytujú prekladateľovi rôzne možnosti, ako sa zachovať v prípade neúplnej zhody záznamov s originálom. Neúplná zhoda v tomto prípade znamená, že sa vety na seba podobajú s výnimkou napríklad jedného rozdielneho slova. Bližšie túto problematiku skúma predovšetkým Hodász a Lagoudaki v (Hodász, et al., 2005) a (Lagoudaki, 2008).

Obaja uvádzajú, že väčšina produktov umožňuje hľadať tzv. čiastočnú zhodu. Za čiastočnú zhodu sa dajú považovať vety a záznamy, ktoré sa líšia jedným slovom, prípadne sú rovnaké minimálne na 75%. Prekladateľ vďaka návrhu s čiastočnou zhodou vo výsledku preloží iba jedno až dve slová miesto celej vety.

Pomerne častá funkcia existujúcich produktov je podľa Hodásza i podpora neprekladaných a nepreložiteľných výrazov, podpora pre automatizáciu prekladu dátumov, formátovaných čísel a pod.

Pokročilejším spôsobom, ako riešiť neúplnú zhodu, a predovšetkým žiadnu zhodu, je pokus o zostavenie prekladu na základe čiastočných nálezov. V tomto prípade sa prekladový program pokúsi nájsť niekoľko záznamov v prekladovej pamäti, ktoré by spolu dokázali poslúžiť na zostavenie originálnej vety a následne i na zostavenie návrhu na preklad.

V akademickej rovine bolo v (Lagoudaki, 2008) ukázané, ako zostrojiť (nielen na základe prekladových pamätí) ďalšie možnosti zostavovania návrhu na preklad v prípade neúplnej zhody. Medzi hlavné postupy patrí použitie lexikónu, použitie štatistických metód, použitie externého korpusu, atď.

2.3 Použitie prekladových pamätí v praxi

V počiatkoch existencie myšlienky prekladových pamätí, a použitia automatického prekladu k prekladu reálnych dokumentov všeobecne, boli tieto techniky profesionálmi odmietané a zavrhované. Tento negatívny postoj bol pravdepodobne spôsobený hlavne tým, že o automatickom preklade sa v jeho počiatkoch

predpokladalo, že dosiahne úrovne profesionálneho prekladu, a ohrozí či úplne vytlačí človeka z pozície prekladateľa.

To sa však nestalo a i keď automatický preklad sa počas posledných desaťročí posunul výrazne napred, jeho úloha dnes nie je konkurovať človeku ako prekladateľovi. Práve naopak. Hlavnou úlohou automatického prekladu, a predovšetkým prekladu použitím prekladových pamätí, je poskytnúť človeku nástroj, ktorý efektívne zautomatizuje manuálnu prácu prekladateľa na preklade tam, kde je reálna kreativita prekladateľa nízka; ponechá mu však priestor na realizáciu a kreativitu tam, kde je kreativita očakávaná, a tiež tam, kde sa automatický preklad prekladu ľudskému zatiaľ nevyrovná.

Objem, v ktorom k reálnemu zefektívneniu dochádza, sa obvykle líši predovšetkým podľa typu dokumentu a jeho zamerania, resp. miery odbornosti. Medzi najviac obľúbené sféry využitia prekladových pamätí patrí pravdepodobne lokalizácia software a preklad odborných textov a dokumentov. Naopak najmenej zaujímavá je (z pohľadu prekladových pamätí) pravdepodobne sféra prekladu beletrie a pod.

2.3.1 Lokalizácia produktu a jeho dokumentácie

V prípade dokumentov, ktoré prechádzajú postupom času rôznymi podobnými verziami (napr. vyššie spomínaná lokalizácia a dokumentácia produktov), je obvyklé a efektívne uchovávať prekladovú pamäť viazanú výhradne k stanovenej sade dokumentov, napríklad výhradne pre konkrétny produkt a jeho dokumentáciu. Prekladová pamäť je silne viazaná k špecifickému obsahu daného produktu, ľahko sa v nej udržuje terminológia, štýl a tón reči. Takáto prekladová pamäť je navyše z pohľadu právneho (z pohľadu autorských práv) ľahšie spravovateľná.

Pokiaľ prekladateľ použitím prekladových pamätí v skutočnosti nevytvára výsledné dokumenty, ale modifikuje samotnú prekladovú pamäť tak, aby z originálu bolo možné priamo vygenerovať výsledok bez ďalšieho zásahu prekladateľa, je takáto prekladová pamäť v podstate samotným dielom.[1] Zároveň si prekladová pamäť

[1] Z pohľadu autorských práv môže i nemusí byť takáto prekladová pamäť braná rovnako, ako výsledný dokument. Zatiaľ neexistuje konsenzus vo veci vlastníctva prekladových pamätí a čaká sa na prvý reálny súdny spor, ktorý pravdepodobne vytvorí precedens pre ďalšie podobné spory. (Smith, 2008)

uchováva vysokú mieru efektivity pri budúcom preklade novej verzie, pretože miera prípadnej modifikácie originálu v ďalšej verzií sa pomerne priamo premieta sa mieru nutnej modifikácie prekladovej pamäte do tejto verzie.[1]

2.3.2 Preklad neverziovaných dokumentov

I keď prístup k automatizácií prekladu pomocou prekladových pamätí je pravdepodobne najúčinnejší, keď je použitý práve na lokalizáciu dokumentov prechádzajúcich postupne viacerými verziami, jeho nasadenie býva obľúbené a užitočné i v ostatných prípadoch, a to predovšetkým pri preklade odborných dokumentov.

Odborné, a i keď neverziované, dokumenty totiž síce neposkytujú možnosť priamo využiť predošlú verziu, zvyknú však používať podobný štýl a tón reči. Majú často i podobné zameranie a terminológiu. Príkladom môžu byť návody k bielej technike, alebo analýzy a posudky týkajúce sa ekonomiky a politiky.

Pri preklade dokumentov, ku ktorým neexistuje priama predošlá verzia, sa prekladateľ spolieha predovšetkým na to, že rôzne dokumenty sú si navzájom do istej miery podobné a existuje v nich pre neho zaujímavé množstvo rovnakých alebo podobných viet.

Efektivita navrhovania možného prekladu je tu však výrazne viac závislá na množstve a podobnosti v minulosti prekladaných dokumentov, t.j. pokiaľ sa prekladateľ v minulosti dlhodobo venoval prekladu návodov k vysávačom, jeho snaha bude v budúcnosti odmenená pravdepodobne hlavne pri preklade ďalšieho podobného návodu k vysávaču.[2]

Predovšetkým tu, v oblasti neverziovaných dokumentov je podstatné, aby produkt používaný k prekladu poskytoval čo najväčšie možnosti práve v prípadoch neúplnej

---

[1] Túto stratégiu konkrétnych prekladových pamätí viazaných na jednotlivé produkty úspešne používajú rôzne veľké spoločnosti (Microsoft, Adobe) a ich klienti – jednotlivý prekladatelia. (Flournoy, et al., 2009) (Richardson, 2007)

[2] Všeobecne je však prax odlišná. Prekladatelia majú tendenciu striedať okruhy a zamerania prekladaných dokumentov. Dôvodom môže byť jednak to, aby spestrili svoju prácu, a jednak dôsledok honby za klientmi – prekladateľ sa častejšie chytá možností podľa ich lukratívnosti než podľa ich zamerania. (Lagoudaki, 2008)

zhody. Dokumenty sa na seba totiž síce podobajú, avšak vo všeobecnom prípade zvyknú zdieľať pomerne málo úplne rovnakých viet. Zostavovanie návrhu z viacerých záznamov sa môže pri väčšej diverzite dokumentov stať priamo kľúčovým nástrojom prekladového produktu.

Samotné zostavenie návrhu je pritom netriviálnou úlohou a kvalita zostaveného návrhu býva hlavným faktorom užitočnosti: Zle zostavený návrh je podobne alebo dokonca menej užitočný ako žiaden návrh – prekladateľa zbytočne zdrží analýza toho, do akej miery je návrh použiteľný, pred tým, než ho celý zmaže a napíše vlastný preklad. Na druhú stranu, i zlý návrh môže prekladateľovi úspešne poslúžiť ako zdroj indícií predtým, než začne písať svoj vlastný preklad.[1]

Navzdory vysokému dopytu a konkurencií v oblasti produktov pracujúcich s prekladovými pamäťami, obvyklé funkcie týchto produktov žiaľ málokedy prekračujú hranicu vyhľadávania čiastočnej zhody, podporu terminológie, dátumov a čísel, ako zhrňuje Hodász v (Hodász, et al., 2005).

[1] Rôzny prekladatelia majú rozdielne reakcie na prakticky rovnakú situáciu. Rozdiel býva sprevádzaný množstvom skúseností v praxi, ale i ďalšími faktormi zahrňujúcimi osobné preferencie a zvyky pri práci. (Lagoudaki, 2008)

3 Chyby a komplikácie prekladu

Problematikou konkrétnych chýb a komplikácií pri preklade sa zaoberajú napríklad články (Abekawa, et al., 2008), (Macklovitch, 1995), (Lagoudaki, 2008), (Hodász, et al., 2005), prípadne i (Jutras, 2000). V tejto kapitole si zhrnieme ich poznatky.

Vo všeobecnom prípade majú prekladatelia tendenciu vytvárať najprv iba predbežný preklad. Tento predbežný preklad je následne upravovaný, často i niekoľko krát, do finálnej podoby. Často tiež platí, že kým predbežný preklad je dielom menej skúsených prekladateľov, a sú k nemu častejšie využité metódy automatického prekladu a výhody prekladových pamätí, finálna verzia prekladu vyžaduje obvykle väčšie skúsenosti a býva preto upravovaná skúsenejšími kolegami, a v tomto prípade už spravidla bez použitia prekladových pamätí či iných automatických nástrojov.[1] Hlavnou úlohou týchto úprav je identifikácia a následné odstránenie chýb, ktoré pri predbežnom preklade vznikajú.

Medzi chyby, ktoré prekladové pamäte v bežnom prípade priamo nezaviňujú, môžeme zaradiť zlý slovosled, príliš doslovný preklad, zanášanie cudzích slov do prekladu, preklepy a spisovnosť, nesprávne použitie podobných slov, čo sú časté chyby hlavne menej skúsených prekladateľov.

Medzi chyby, ktoré vznikajú predovšetkým práve použitím prekladových pamätí, môžeme zaradiť chyby ako náhla zmena štýlu alebo tónu reči, odchýlky od dohodnutej terminológie, nesprávny preklad v prípade dvojzmyselnosti vety z originálu, a podobne.

Ďalším zaujímavým typom chýb sú preklepy v nepreložiteľných častiach textu – v číslach, rokoch, dátumoch, adresách, matematických výrazoch a pod. Tieto chyby môžu unikať pozornosti i pri opakovanej revízií prekladu, pretože sa jedná o časti textu, ktoré prekladateľ priamo neprekladá a podvedome často úplne ignoruje a preskakuje.

[1] Je nutné poznamenať, že v prípade lokalizácie produktov často neexistuje „predbežný preklad“, ale „predbežná prekladová pamäť“, ktorou bude výsledný „preložený dokument“ vygenerovaný. V tomto prípade sú opravy vykonávané priamo na prekladovej pamäti.

Existujú i ďalšie typy chýb v preklade, mimo iné napríklad neúplnosť prekladu. Bez ohľadu na skúsenosti prekladateľa sa môže stať, že prekladateľ neúmyselne zabudne preložiť celú vetu, odsek, či dokonca stránku. Z pohľadu prekladových pamätí tento typ chýb nie je relevantný, avšak vynechanie časti vety alebo dôležitého slova nás eventuálne zaujímať môže.

3.1 Štýl a tón reči

Vzhľadom k tomu, že vo veľkom množstve prípadov býva prekladová pamäť výsledkom mnohoročnej práce prekladateľa na rôznych dokumentoch, mení sa často zameranie a prípadne i okruh dokumentov, tón a štýl používanej reči v týchto dokumentoch, použitie pasívnej alebo aktívnej formy, a ďalšie identifikovateľné atribúty.

Prekladateľovi sú pri preklade oddelene predkladané návrhy zostavené z rôznych záznamov v prekladovej pamäti. I keď jednotlivé vety prekladateľ prekladá postupne (a každá z nich je sama o sebe obvykle konzistentná svojim štýlom), menej skúsený prekladateľ môže ľahko strácať kontext a súvis s okolitým textom, a vytvárať tak zlátaninu – dokument, ktorý vo výsledku nepôsobí ani plynulo ani profesionálne.

3.2 Dohodnutá terminológia

I v prípade, že prekladateľ sa zameriava na konkrétnu oblasť klientov, venuje sa úzkemu výberu okruhov a udržuje rovnaký štýl a tón reči v priebehu času, rôzni klienti obvykle trvajú na rôznej terminológií. Často pritom ide o drobné rozdiely v slovách ako napríklad „klikať“ vs. „klepať“ myšou, „užívateľ“ vs. „používateľ“, alebo anglické „cursor“ vs. „caret“. Prekladové pamäte tak môžu obsahovať významovo rovnaké vety, obsahujúce však rôznu terminológiu.

Chyba v terminológií navyše nemusí nutne poukazovať na malé skúsenosti prekladateľa, a výsledný dokument môže pre bežného čitateľa znieť plynulo a bezchybne, a predsa pritom nedodržovať dohodnutú terminológiu. Špeciálne v prípade, kedy je dokument príliš dlhý a času na preklad je málo, býva prekladateľov viac. Dokument si jednotlivý prekladatelia rozdelia na menšie časti a každý prekladá

svoju časť samostatne. Na koniec tieto časti jeden z prekladateľov spojí dohromady do výsledného dokumentu. Tu je viac než nevyhnutné, aby sa vopred stanovili jednoznačné pravidlá, keďže preklady jednotlivých častí vznikajú oddelene a „s použitím rôznych názorov na preklad“.[1]

Medzi obzvlášť nepríjemné patrí náhla zmena dohodnutej terminológie zo strany zadávateľa. Takáto zmena sa obvykle týka konkrétnych slov, či už s cieľom zjednotiť doteraz nešpecifikovanú terminológiu, alebo prispôsobiť sa vonkajšej zmene v jazyku.[2]

Čiastočná podpora spravovania terminológie, síce býva zahrnutá v známejších komerčných produktoch, jedná sa však obvykle iba o jednoduché moduly, ktoré spravidla neprekračujú hranice jednoduchého „nájdi a nahraď“.

3.3 Dvojzmyselnosť

Medzi chyby, zanesené do prekladu použitím prekladových pamätí (ale i použitím metód automatického prekladu všeobecne), môžeme zaradiť aj nesprávny preklad viaczmyselných viet. Pod viaczmyselnou vetou si predstavujeme vetu, ktorá v jazyku originálu znamená v rôznom kontexte rôzne veci. V cieľovom jazyku majú naopak rôzne významy tejto vety (veľmi) odlišné znenie.[3]

Rovnako ako v predošlých prípadoch, užívateľské prostredie produktov pracujúcich s prekladovými pamäťami môže prekladateľa viesť k momentálnej strate kontextu. Veta, ktorá má v kontexte originálu jednoznačný význam, ale sama o sebe je viaczmyselná, môže byť v prekladovej pamäti nesprávne vyhľadaná (prípadne

---

[1] V praxi je bežné, že prekladateľ (prípadne celý tým prekladateľov) pracujúci na preklade dostane od zadávateľa inštrukcie k prekladu obsahujúce mimo iné aj špecifikáciu terminológie. Tá má za úlohu zaručiť, že výsledný dokument bude konzistentný. Žiaľ, nie je neobvyklé, že prekladatelia tieto inštrukcie (často nechtiac) ignorujú alebo sa k nim inštrukcie nedostanú včas či vôbec. Nekonzistentný výsledok je bežnou realitou, čím väčší projekt, tým viac.

[2] Ako príklad môže poslúžiť práve zmena z „klikať“ na „klepať“ myšou (v dokumentácií nemenovanej sady produktov istej veľkej spoločnosti).

[3] Ako príklad môže poslúžiť veta „Only then he realized the plan.“ Táto totiž môže znamenať v preklade „Až vtedy si uvedomil, čo je v pláne.“, ale zároveň i „Až potom uskutočnil plán.“ O tom, ktorý z významov je správny v tomto prípade napovie pravdepodobne iba kontext. Viď príklady ku kapitole Cudzie slová a podobné slová na strane 16.

i zostrojená z viacerých zdrojových záznamov). Prekladateľovi je navrhnutý nevhodný preklad, ktorý však môže byť jednak gramaticky správne, a jednak môže pôsobiť „správnym dojmom“ – pri príliš doslovnom preklade vety a synchronizovanom preklade významu jednotlivých slov nemusí byť prekladateľovi na prvý pohľad jasné, že výsledná veta v skutočnosti v cieľovom jazyku a v danom kontexte nemá reálny význam – menej skúsený prekladateľ totiž danej vete „rozumie“, hlavne pokiaľ má zároveň k dispozícii aj viaczmyselný originál.

3.4 Čísla, adresy a matematické výrazy

I keď je pravdou, že mnohé produkty používajúce prekladové pamäte v sebe obsahujú moduly podpory neprekladaných výrazov, akými sú čísla, roky, dátumy, adresy, matematické výrazy a pod.; vo všeobecnosti to pravda byť nemusí. Kontrola týchto častí textu, pokiaľ správnosť nie je priamo zaručená prekladovým produktom, je pre prekladateľa veľmi nudná činnosť, pretože ako podotýka Macklovitch v (Macklovitch, 1995), „množstvo intelektu ku kontrole potrebné je prakticky nulové“.

3.5 Chýbajúce vety a slová

Chýbajúce vety, odseky, či celé stránky textu sú žiaľ bežnou a navyše veľmi nepríjemnou chybou manuálneho prekladu. Dôvodom k tomu býva vyčerpanosť, nepozornosť, ale i nehody, ako napríklad nechcené stlačenie nevhodnej klávesy v editore či prekladovom produkte.

Použitie automatického prekladu tento problém z časti rieši samo od seba, pretože prekladové produkty obvykle prekladateľa priamo vyzývajú k prekladu zatiaľ nepreložených viet. Už však nezaručia prípadné chýbajúce úseky a dôležité slová v rámci jednotlivých viet.

Veta, resp. zmysluplná časť vety je pre prekladový produkt obvykle ďalej nedeliteľná časť textu. Prekladový produkt z pravidla neoveruje, či každé „dôležité“ slovo z originálu má svoj ekvivalent v preklade. Vo vete ako „Zvoľte si bielu, modrú alebo červenú.“ je možné, že prekladateľ jednu z možností nedopatrením vynechá.

3.6 Slovosled, preklepy

Zlý slovosled a príliš doslovný preklad sa dajú charakterizovať ako problémy plynulosti výsledného prekladu a sú obvykle známkou profesionality prekladateľa. Ich odhalenie je pritom najrýchlejšie a najpravdepodobnejšie, pokiaľ čitateľ nemá vôbec originál k dispozícií. I neskúsený čitateľ je schopný povedať, že niečo nie je v poriadku. Výsledkom môže byť často komicky znejúci text alebo nezrozumiteľný až mätúci text.

3.7 Cudzie slová a podobné slová

Použitie (v tomto prípade zneužitie) cudzích slov je tiež často otázkou profesionality a v istých kruhoch nemusí byť dokonca ani zrejmé a nápadné, pokiaľ je v danej oblasti zvykom používať termíny z oboch jazykov. Avšak striedanie týchto slov v jednom dokumente je z hľadiska profesionálneho prekladateľa neprípustné. Príkladom môže byť použitie slov „software“ a „softvér“ v prostredí informatiky. Ďalšie časté príklady zahrňujú dnes obľúbené zapožičané termíny ako „meeting“ alebo „briefing“.

Špeciálnou kategóriou sú slová, ktoré znejú v oboch jazykoch podobne, majú však rôzny význam. Jednoduchým príkladom môže byť anglické „definitely“ („rozhodne“, „určite“, „samozrejme“) voči slovu „definitívne“ („once for all”). I keď majú tieto slová podobné znenie, ich význam je rozdielny.

Ešte zaujímavejším príkladom môže byť už spomínané slovo „realizovať“ voči anglickému „realize“. V skutočnosti sú ďaleko častejšími prekladmi k slovu „realizovať“ preklady „execute“, „carry out“, „implement“ alebo „put into effect“. A naopak k slovu „realize“ sú ďaleko častejšími preklady „uvedomiť si“, „pochopiť“, „predstaviť si“ alebo „získať“. Situácia je však zložitejšia, pretože v niektorých kontextoch môže byť preklad slov „realizovať“ na „realize“ skutočne správny („to realize a plan or an idea“ vs. „realizovať plán či nápad“) aj napriek tomu, že vo väčšine prípadov sa jedná o chybu („to realize a situation“ vs. „uvedomiť si situáciu“).

4 Aktuálny výskum a existujúce riešenia

Hlavné odvetvia, ktoré sa snažia hľadať a odstrániť komplikácie, slabiny a chyby automatického prekladu použitím prekladových pamätí, sa dajú rozdeliť primárne do niekoľkých kategórií, ktorých nasadenie sa navyše navzájom nevylučuje.

Prvou kategóriou je hľadanie riešenia zásahom do originálu dokumentu – zavedením pravidiel a zásad písania už pri vzniku originálnych dokumentov.[1]

Druhou kategóriou je hľadanie efektívnejších a spoľahlivejších metód automatického prekladu; výskumom a implementáciou nových možností, ako zostaviť adekvátny a viac spoľahlivý návrh na preklad.

Treťou kategóriou je odhaľovanie chýb vo výslednom preklade a upozorňovanie na ne; skúmaním a identifikáciou najčastejších chýb prekladateľov a vytváranie pravidiel na ich odhaľovanie.

I keď sa na prvý pohľad zdá, že druhá a tretia kategória skúma ten istý problém, nie je to v skutočnosti pravda ani implementačne, ani metodologicky. Od automatického prekladu sa totiž očakáva, že bude navrhovať možný preklad, a že to bude robiť pre každý kus zdrojového textu, t.j. i v prípade, že si so zdrojovým textom nebude vedieť poradiť. Či už bude výsledok automatického prekladu korektný, alebo nie, úlohou automatického prekladu je tento výsledok poskytnúť v každom prípade.

Naopak kontrola po automatickom (ale i manuálnom) preklade sa k problému stavia z opačnej strany. Výsledok prekladu dostane k dispozícii a jej úlohou je tento výsledok zanalyzovať na základe znalosti jazyka. V prípade, že si so zdrojovým textom nebude vedieť poradiť, nemusí robiť vôbec nič.[2] Ako píše Macklovitch v (Macklovitch, 1995), pre prekladateľa je vo všeobecnom prípade vždy lepšie mať k dispozícii aspoň nejaký (nedokonalý) nástroj na kontrolu, ako žiaden.

---

[1] Tomuto spôsobu sa obvykle hovorí riadené písanie.

[2] Predovšetkým pokiaľ je podozrenie, že modul kontroly nerozumie jazyku v dostatočnej miere, je podstatné, aby takáto kontrola nehlásila užívateľovi svoje zlyhanie. V opačnom prípade by sa totiž stratila jej primárna úloha – ušetriť prekladateľov čas a zefektívniť jeho prácu.

Ďalším rozdielom je, že nástroj na automatický preklad sa dá použiť spravidla iba jeden, ale nástrojov na kontrolu prekladu je možné použiť viac. Automatický preklad musí vziať do úvahy jazyk ako celok a vyrovnať sa s ním po všetkých stránkach, výnimky nevynímajúc, v jednom „veľkom algoritme“.[1] Naopak kontrola prekladu si môže problém hravo rozdeliť na rôzne typy a každý typ riešiť samostatne. Každá časť si stanoví konkrétnu oblasť záujmu a k svojmu cieľu si zvolí najvhodnejšie prostriedky a metódy. Jednotlivé časti medzi sebou nemusia byť vôbec kompatibilné a nemusia nijak spolupracovať.

4.1 Riešenie „Global English“

Prvou kategóriou, pravdepodobne najpriamočiarejšou, ale zároveň problematicky použiteľnou, je „zmena“ jazyka originálu. Microsoft napríklad používa sadu pravidiel a zásad pri písaní originálnej dokumentácie, ktorú nazýva „Global English“. Výberom z týchto pravidiel sú napríklad používanie jednoduchých viet miesto súvetí, alebo delenie dlhých viet do odrážkových zoznamov. (Richardson, 2007)

Ako uvádza Richardson vo svojej prezentácií, Microsoft využíva túto stratégiu úspešne vo svojich produktoch už niekoľko rokov. I keď tento prístup neodstraňuje nutne chyby automatického prekladu použitím prekladových pamätí, výrazne ovplyvňuje mieru, do akej prekladové pamäte navrhujú korektný a prijateľný preklad. Richardson uvádza, že spokojnosť koncového užívateľa s automatickým prekladom (bez akéhokoľvek zásahu človeka) pri dodržaní pravidiel „Global English“ je obdobná spokojnosti s prekladom, v ktorom človek – prekladateľ zohral hlavnú úlohu.

Očividnou nevýhodou „Global English“ je však nutnosť meniť štýl a formu originálu, čo je vo všeobecnom prípade nevýhodné, nepoužiteľné až neprípustné.

[1] Nech už je tento „veľký algoritmus“ poskladaný z čohokoľvek, musí existovať jasný spôsob, ako jednotlivé prvky poskladať dohromady. V prípade viacerých výsledkov musí existovať spôsob, ako rozhodnúť, ktorý výsledok je najlepší. Použitie a kombinácia rôznych techník (štatistické metódy, kognitívne metódy, pravidlá jazyka, hľadanie vzorov, lexikóny, prekladové pamäte) tak predstavuje samostatný problém.

4.2 Párovanie a lingvistické informácie

Druhou kategóriou prístupu k zlepšeniu výsledkov prekladových pamätí je snaha zlepšiť vyhľadávanie správneho navrhovaného prekladu a snaha zostrojiť presnejší a korektnejší preklad pri neúplnej zhode. Tento prístup rovnako neodstraňuje priamo chyby automatického prekladu použitím prekladových pamätí, ovplyvňuje však mieru, do akej prekladové pamäte navrhujú gramaticky korektný preklad.

K tomuto cieľu sú, momentálne zdá sa iba na úrovni výskumu a experimentov, do prekladovej pamäte zahrnuté napríklad moduly poskytujúce hľadanie zarovnania a párovania na úrovni slov a doplnenie slov o lingvistické informácie. Tieto dopĺňajú bežné vyhľadávanie o nové informácie, a predovšetkým pri neúplnej zhode ponúkajú k dispozícií prostriedky nevyhnutné k zostaveniu gramaticky korektného návrhu. (Hodász, et al., 2005) (Hua, et al., 2005)

4.3 Hľadanie chýb v preklade

Treťou, kategóriou prístupu je priamo hľadanie chýb v preklade. Táto kategória si berie za cieľ dodatočne upozorniť prekladateľa na nesprávny či zvláštny preklad, nedodržanie terminológie, náhlu zmenu tónu či štýlu, či opomenutie častí viet i celých odsekov a stránok; a prípadne mu navrhnúť i možnú opravu.

Ako už bolo spomenuté, v bežnom prípade produkujú profesionálni prekladatelia najskôr predbežnú verziu prekladu, ktorú potom neskôr upravujú, často i viackrát, do finálnej podoby.

Toto však pri „neprofesionálnom“ preklade, napríklad dobrovoľníkmi, už nie nutne platí, pretože dobrovoľníci prekladu venujú výrazne menej času, a motivácia k prekladu (a k jeho kvalite) má často úplne odlišný základ. Do role vstupuje pravdepodobne i fakt, že dobrovoľníci bývajú menej skúsení.

Hľadanie chýb v preklade tak nemusí byť iba otázkou zefektívnenia úprav medzi predbežnou a finálnou verziou. Naopak, pokiaľ hľadanie chýb nie je stopercentné (a popravde, zatiaľ nie sme k stopercentnej kontrole nijak blízko), je prekladateľ nútený celý predbežný preklad preštudovať tak či onak. Výhodou automatickej

kontroly je však to, že viac-menej zvýši kvalitu výsledného dokumentu – zníži totiž počet chýb, ktoré by (i skúsený) prekladateľ mohol pri kontrole eventuálne prehliadnuť.

4.3.1 Existujúce riešenia pre iné jazyky

Práve z týchto vyššie uvedených dôvodov by bolo veľmi pozitívnym prínosom, ak by prekladové produkty dokázali prekladateľa včas upozorniť na chyby, ktoré pri preklade zvykne robiť – kvalita predbežného i dobrovoľníckeho prekladu by sa zvýšila.

Pozrime sa na už existujúce riešenia, ktoré existujú napríklad pre japončinu či francúzštinu a zhrňme si, aké chyby v preklade sa pokúšajú hľadať a s akou úspešnosťou výsledkov sa stretávajú.

4.3.1.1 Z angličtiny do japončiny

Aby bolo možné takýto systém vyvinúť, Abekawa a Kageura sa v (Abekawa, et al., 2008) pokúsili zamerať na dve otázky:

- aké sú základné typy zmien, ktoré prekladatelia robia pri úprave predbežných prekladov do finálnych prekladov; a
- čo u prekladateľa signalizuje, že preklad nie je v poriadku a je nutné ho upraviť.

Zo svojho výskumu určili, že skúsený prekladateľ pri úprave predbežného prekladu skúma a (často podvedome) identifikuje tzv. *zlé stavy* v preklade a k nim ich príčinu. Medzi tieto *stavy* zaradili zlý preklad, nezrozumitelnosť prekladu, neprirodzenosť prekladu, nevhodnosť tónu a štýlu reči, a prehrešky voči redakčným zásadám.[1]

[1] Abekawa a Kageura sa vo svojej práci venovali výhradne prekladu (a tam vznikajúcim problémom) z angličtiny do japončiny. I keď rôzne dvojice jazykov zvyknú produkovať spravidla trochu odlišnú sadu problémov (i obrátenie dvojice jazykov je z tohto pohľadu iná dvojica), zdá sa, že je možné eventuelne zovšeobecniť a zostaviť „nadmnožinu“ týchto problémov.

Tieto zlé stavy a ich príčiny ďalej rozviedli a vytvorili z nich sadu signalizujúcich pravidiel, ktoré tieto stavy pokrývali.[1] Výsledné pravidlá rozdelili do piatich kategórií podľa typu rysov jazyka v nich zachytených a následne sa pokúsili o experiment, ktorý by ukázal, do akej miery sú tieto pravidlá úspešné:

- Rysy jazyka týkajúce sa čitateľnosti angličtiny. Komplexnosť angličtiny totiž môže výrazne ovplyvniť kvalitu predbežného prekladu. Tu sa zamerali na počítateľné veci ako dĺžka slov, počet slov, počet slovies vo vete, počet čiarok a pod.
- Rysy odzrkadľujúce korešpondenciu medzi anglickým textom a japonským textom. U japončiny, ktorá má odlišnú štruktúru, je príliš doslovný preklad často i prekladom neprirodzeným a zvláštnym. Na druhú stranu, vysoká miera podobnosti poradia niektorých slov môže byť pozitívnym znakom, pretože nasvedčuje tomu, že je v preklade zachovaný správny tok informácie.
- Rysy jazyka týkajúce sa japonských slovies. Charakteristika použitia slovies v japončine je viazaná na prostredie v ktorom sa nachádzajú.
- Rysy jazyka týkajúce sa prirodzenosti japonského textu. Tu sa jedná predovšetkým o opakovania a nadbytočné použitie slov, ktoré môžu viesť k neprirodzenému prejavu.
- Rysy jazyka týkajúce sa komplexnosti japončiny. Pokiaľ je preklad nadmerne komplexný, môže byť ťažko čitateľný a zmätočný. Komplexnosť štruktúry viet, dĺžka viet a počet čiarok môžu byť použité ako pravidlá zobrazujúce komplexnosť textu.

Vo výsledkoch Abekawa a Kageura uvádzajú vysokú mieru úspešnosti svojho pokusu, a ďalej študujú prípady, kde experiment označil správne vety za nesprávne a naopak, nesprávne vety za spávne.

[1] Abekawa a Kageura mimo iné argumentuje, že prekladateľ pri hľadaní stavov využíva intuíciu a skúsenosti, a je schopný sa s problémom vyrovnať na základe predošlých skúseností. Počítače pochopiteľne tohto nie sú schopné a tak je nutné množinu stavov previesť do „computationally tractable triggers“ – čiže sadu pravidiel, ktoré počítač dokáže správne „spočítať“ a použiť.

V prípade japončiny za najviac prínosné považujú pravidlá týkajúce sa japonských slovies. Ako dôvod uvádzajú to, že v japončine sú prípony u slovies viazané na tón reči – rôzne prípony slovesa nemenia význam vety, ale jej tón. Naopak za najmenej prínosné považujú pravidlá týkajúce sa čitateľnosti angličtiny a komplexnosti japončiny.

Do tretice uvádzajú, že rysy jazyka týkajúce sa prirodzenosti japonského textu (opakovanie a nadbytočné použitie slov) síce zaznamenali najmenej využitia (t.j. signalizovali najmenej krát), zato ich presnosť a spoľahlivosť bola najvyššia (t.j. keď potrebu zmeny signalizovalo pravidlo z tejto kategórie, zmena bola skutočne potrebná).

4.3.1.2 Z angličtiny do francúzštiny

Zaujímavým a existujúcim nástrojom na hľadanie chýb v preklade je nástroj TransCheck, bližšie popisovaný v (Macklovitch, 1995) a (Jutras, 2000). Tento si vzal za úlohu hľadanie chýb pri preklade z angličtiny do francúzštiny. Macklovitch, a neskôr Jutras, zvolili k problému úplne iný prístup. Rozdelili si úlohu kontroly prekladu na niekoľko menších logických podčastí a každej z nich sa venovali zvlášť:

- Illicit borrowings – zakázané použitie anglických slov vo francúzskom texte. Jedná sa predovšetkým o anglické slová ako „brunch“ alebo „briefing“, ktoré sú podľa všeobecného zaužívaného štandardu nevhodné pre písaný francúzsky text. Vzhľadom na podobnosť angličtiny a francúzštiny je však použitie anglických slov vo francúzštine obvyklým javom.
- Deceptive cognates (tzv. faux amis) – mylné použitie podobne znejúcich slov s rozdielnym významom. Ako príklad uvádzajú kombináciu anglického „library“ (knižnica) voči francúzskemu „librairie“ (kníhkupectvo).
- Terminológia – miešanie rôznych termínov pre rovnaký pojem.
- Chýbajúce vety, odseky, strany – ako dôsledok nepozornosti či únavy.
- Čísla, dátumy, matematické výrazy – ako neprekladané, ale rozpoznateľné a porovnateľné časti textu.

Zakiaľ pri rozpoznávaní zakázaných slov (illicit borrowings) úlohu považovali za jednoduchú, pri rozpoznávaní podobne znejúcich slov (deceptive cognates) zvýraznili, že úloha nie je zďaleka triviálna. Tieto slová a výrazy sú totiž súčasťou slovníka i gramatiky daného jazyka. Problém použitia nie je v zasadení slov do viet, ale v zasadení do kontextu. Navrhujú riešenie, ktoré na úrovni odsekov a viet páruje jednotlivé slová, a následne v týchto pároch hľadá zakázané dvojice (ako napríklad „library" vs. „librairie"). Zároveň však poukazujú na to, že táto metóda je závislá na kvalite správneho spárovania odsekov a viet[1] a následne na kvalite správneho spárovania jednotlivých slov. Aby bola situácia ešte zložitejšia, Macklovitch uvádza príklady toho, kedy tieto deceptive cognates nie sú v skutočnosti deceptive cognates v závislosti na kontexte – kým v jednom prípade sa jedná o nesprávny preklad, v inom kontexte je preklad úplne v poriadku (anglické „to graduate" vs. francúzske „graduer", ktoré znamená to isté v zmysle postupného zvyšovania, ale nie je v zmysle úspešného ukončenia štúdia).

Pri riešení kontroly konzistencie terminológie poukazujú na to, že problém, ktorý sa na prvý pohľad zdá triviálny (hľadanie predpísaných párov termínov v oboch jazykoch), má svoje úskalia nielen v podobe morfológie jazykov, ale napríklad i v podobe následného skracovania termínov. Ako príklad uvádza výraz „tireur d'élite" (ostreľovač, anglicky jednoslovne „sniper"), oprávnene skracovaný na „tireur" po tom, čo je v texte prvý krát použitý úplný názov pojmu. Uvádza, že nie je možné túto dvojitú možnosť nezahrnúť (vylúčiť skrátené „tireur" zo správnej terminológie, čo by viedlo k nadmernej mierne falošných poplachov); ale ani jednoducho zahrnúť (povoliť skrátené „tireur" kdekoľvek, čo by viedlo k riziku zlyhania detekcie pri prvom použití) do terminologickej databáze.

Pri hľadaní chýbajúcich odsekov, viet, či dokonca častí viet, kde spoliehajú na robustnosť postupného párovania odsekov, následne viet, a postupne až na úroveň slov, poukazuje Jutras na problémy, ktoré prinášajú vety ako „It will definitely be done by January 1, 2001." vs. „It will definitely, and I mean definitely, be done by

[1] Správne spárovanie viet býva problematické nielen preto, že preložené vety majú často rôznu dĺžku voči originálu, ale i preto, že jedna veta môže byť preložená do viacerých viet a naopak. Významnú roľu ďalej zohráva i fakt, že prekladateľ môže (netriviálne veľkú) časť textu v preklade nechtiac úplne vynechať.

January 1, 2001." Pre nimi navrhovaný systém sú totiž tieto vety z pohľadu párovania nerozpoznateľné, t.j. obe sú úspešne spárovateľné s francúzskym prekladom prvej z nich.

Na záver skúmajú možnosti kontroly čísel, dátumov a matematických výrazov. Navrhujú, že sa v tomto prípade jedná viac o prepis, transkripciu, než o preklad. Zároveň však uvádzajú príklady toho, kedy číslo „2" a slovo „two" musí byť správne chápané ako ekvivalentný istý výraz. Na druhú stranu poukazujú na to, že slovo „second" v sebe okrem čísla 2 nesie i informáciu o ordinalite. Jutras navrhuje použitie „transformátoru" výrazov, ktorý by unifikoval a normalizoval rôzne tvary pojmov a ponúkal tak porovnávacej vrstve možnosť pozerať na výrazy „second" a „2nd" ako na ten istý normalizovaný pojem, ale odlišný od výrazov „two" a „2".

5 Návrh riešenia

5.1 Z pohľadu jazyka

Základný rozdiel medzi češtinou (resp. slovenčinou) a angličtinou či francúzštinou je z nášho pohľadu v morfológii.[1] Angličtina a francúzština má morfológiu pomerne jednoduchú, a v prípadnom algoritme na kontrolu prekladu zvládne túto morfológiu pomerne dobre vyriešiť napríklad použitie „fuzzy“ vyhľadávania[2], alebo aplikovanie neveľkého počtu jednoduchších pravidiel.

V češtine je tento problém samozrejme trochu významnejší. Obyčajné „fuzzy“ už nestačí, pretože jednotlivé tvary slova sa často líšia o netriviálny počet písmen. „Fuzzy“ prístup navyše neodhalí prípadné chyby v preklepoch. Riešením by bolo zaniesť do problému skutočný morfologický slovník, resp. nástroj, ktorý by dokázal slová viet previesť na ich základný tvar, či naopak.

Pravidlá, ktoré by následne hľadali napríklad zakázané dvojice (viď Illicit borrowings v kapitole 4.3.1.2), by sa tak nemuseli ďalej zaoberať morfológiou. Opačná možnosť je zaviesť morfológiu priamo do pravidiel a naučiť pravidlá rozoznávať všetky tvary hľadaného slova.

5.2 Z pohľadu implementácie

Pred začiatkom práce som si stanovil niekoľko základných princípov, ktoré by nové riešenie malo dosiahnuť a ponúknuť. Vychádzal som pritom hlavne z poznatkov získaných štúdiom problémov kontroly prekladu, a tiež z návrhov profesionálneho prekladateľa. Zároveň som sa pokúsil zamerať na to, že moje riešenie by malo byť v budúcnosti ľahko rozšíriteľné; a jeho jednotlivé časti v prípade potreby voľne zameniteľné či úplne odpojiteľné.

---

[1] Rozdiel je i v syntaxi. Syntax sa však kontroluje veľmi problematicky, preto sa budeme v rámci tejto práce zaoberať iba morfológiou.

[2] Pod pojmom „fuzzy“ vyhľadávanie sa myslí vyhľadávanie, ktoré dovoľuje drobné odlišnosti v slovách. Obvykle je možné u takéhoto vyhľadávania nastaviť, kde sa smú odlišnosti vyskytovať (na začiatku, v strede, či na konci slova); a ako veľké odlišnosti má vyhľadávanie tolerovať (počet jednotlivých písmen, počet odlišných skupín písmen). Príkladom jednoduchého použitia „fuzzy“ vyhľadávania je nástroj MemoryMining.

Moje riešenie by preto malo naplniť a podporovať nasledujúce body:

- Možnosť zostaviť si vlastnú „výrobnú linku“ – jednotlivé časti (moduly) môjho riešenia by mali jasne špecifikovať svoju funkciu, a mali by nasledovať vopred definované jednotné rozhranie tak, aby bolo možné podobné moduly voľne zamieňať v prípade potreby za iné. Napríklad: Konkrétny modul čítania vstupných dát by mal byť zameraný na jeden typ vstupného formátu, ale množina všetkých modulov čítania vstupných dát by mala poskytovať rozhranie dostatočné k implementácií rôznych typov vstupných dát, a taktiež rôznych umiestnení týchto dát.
- Ovplyvniteľné poradie kontroly – moduly určené na úpravu vstupného textu a moduly určené na kontrolu prekladu by mali byť navzájom kombinovateľné podľa aktuálnej potreby. Jednotlivé moduly by sa nemali navzájom vylučovať, ale ani bezpodmienečne vyžadovať, pokiaľ to v konkrétnych prípadoch nevyžaduje priamo ich logická povaha.
- Rozdeliteľnosť pravidiel – ktorýkoľvek kontrolný modul by mal byť použiteľný viackrát, aby bolo možné oddelene definovať viaceré súbory pravidiel pre ten istý modul (napríklad súbor globálne zakázaných fráz, a súbor zakázaných fráz pre daný projekt).
- Samostatnosť – vzhľadom na predpoklad, že kontrola skutočnej profesionálnej prekladovej pamäte môže pri veľkom množstve pravidiel byť výpočtovo náročná, toto riešenie by nemalo vyžadovať priebežný vstup od užívateľa. Užívateľ by mal mať možnosť špecifikovať všetky potrebné informácie pred spustením kontroly. Kontrola by potom mala byť schopná prebiehať v tichom móde.
- I keď výsledky jednotlivých kontrolných modulov môžu mať rozdielnu povahu, mali by byť užívateľovi ponúknuté rovnakým spôsobom. Užívateľ by nemal dôjsť k záveru, že kombinovanie rôznych typov pravidiel či modulov robí prácu s aplikáciou zložitejšou a náročnejšou.

5.3 Typy hľadaných chýb

Z konzultácie problému s profesionálnym prekladateľom a tiež z informácií a ideí získaných štúdiom už existujúcich riešení som sa pre začiatok rozhodol zamerať sa predovšetkým na tieto najpálčivejšie[1] úlohy:

- Overovanie doporučených prekladov slov a fráz.
- Kontrolovanie stanovenej terminológie.
- Kontrolovanie zakázaných prekladov slov a fráz.
- Kontrolovanie notoricky známych nevhodných výrazov obecne.
- Kontrolovanie zabudnutých, zmazaných, či nepreložených záznamov.
- Experimentovanie s dĺžkou vety a počtom slov.[2]

5.4 Typy vyhľadávacích modulov

Vybrané najpálčivejšie úlohy by sa z pohľadu navrhnutého riešenia mali dať spojiť do nasledujúcich modulov:

- Presadzovanie konkrétneho prekladu originálu (zahrňuje doporučené frázy a terminológiu). Tento modul by mal používať vyhľadávacie pravidlá, podľa ktorých najskôr overí aplikovateľnosť jednotlivého pravidla (t.j. zistí, či sa v zdrojovom texte nachádza slovo alebo fráza, na ktorú dané pravidlo reaguje), a pokiaľ áno, následne overí, či záznam toto pravidlo splňuje.
- Zakazovanie konkrétneho prekladu a konkrétnych fráz (zahrňuje zakázané preklady a notoricky známe nevhodné výrazy). Tento modul, podobne ako predošlý, by mal taktiež používať vyhľadávacie pravidlá

[1] Pre profesionálneho prekladateľa by bolo ideálne naplniť nástroj „pravidlami“ založenými na chybách z minulosti, aby sa u neho rovnaká chyba už viac neopakovala. Navyše skúsenejší prekladateľ by mohol takýto súbor pravidiel poskytnúť ďalej, menej skúseným kolegom. Taktiež, z pragmatického pohľadu videl profesionálny prekladateľ najviac efektívne zamerať sa najskôr na problémy, ktoré sa dajú ľahko algoritmicky popísať, a ktoré k funkčnosti vyžadujú radšej systém písaných pravidiel, než „hromadu textu, na ktorom sa niečo naučia“.

[2] Profesionálny prekladateľ sa zdal byť skeptický voči podobným experimentom. Rovnako sa zdal byť skeptický voči štatistickým metódam, o ktorých by nebol schopný povedať, čo sa deje vo vnútri a prečo to funguje (či nefunguje) správne.

(najlepšie i podobného charakteru a syntaxe), podľa ktorých najskôr overí aplikovateľnosť konkrétneho pravidla, a následne overí prípadnú chybovosť záznamu.

- Porovnávanie dĺžky viet a počtu slov. Tento modul by sa mal pomocou niekoľkých voliteľných čísel pokúsiť hľadať príliš krátke a príliš dlhé preklady vzhľadom na originál. Cieľom je zistiť a overiť, či vety, ktoré sú napríklad kratšie ako polovica originálu nezvyknú byť vety neúplné; a naopak, vety dlhšie ako napríklad dvojnásobok originálu nezvyknú byť vety zbytočne zložité.
- Porovnávanie textového obsahu viet. Tento modul by mal overiť, či sú obe časti záznamu vyplnené, a zároveň či nie sú takmer identické. Cieľom je identifikovať záznamy, kde prekladateľ nechtiac zmazal celú časť záznamu; a záznamy, ktoré prekladateľ vôbec nepreložil. Modul by pri overovaní nemal brať ohľad na drobné rozdiely predovšetkým v interpunkcií a prázdnych znakoch.

5.5 Pravidlá a syntax vstupných dát kontroly

Pravidlá pre moduly, ktoré budú pravidlá vyžadovať a používať, by mali používať podľa možnosti jednoduchú syntax, najlepšie podobnú nejakej už existujúcej a známej syntaxi. Napríklad zástupné symboly „wildcards“ známe zo zápisu v súborových systémoch, prípadne zápis podobný regulárnym výrazom.

Súbory týchto pravidiel a ďalšie vstupné súbory by mali byť založené na jednoduchých textových formátoch ako sú INI a XML, ku ktorým existuje mnoho dobrých editorov, uľahčujúcich prácu s nimi.

6 Architektúra riešenia

Minimalistické jadro výslednej aplikácie obsahuje základné funkcie na načítanie informácií o projekte, nevyhnutnú funkcionalitu medzi-modulového komunikačného rozhrania, a riadenie samotnej kostry výpočtu. Projektom sa tu rozumie aktuálne nastavenie aplikácie a jej jednotlivých častí: zoznam a poradie použitých metód hľadania, súbory pravidiel hľadania pre rôzne metódy, zoznam vstupných súborov, špecifikácia očakávaného výstupu, atď.

Čítanie vstupných dát, úprava vstupných dát, implementácia samotného hľadania chýb, i sprostredkovanie prípadných výsledkov je riešené samostatne pomocou zásuvných modulov. Tieto zásuvné moduly sa delia na tri skupiny a každá z nich definuje kompaktné rozhranie komunikácie s jadrom:

- Vstup – moduly, ktoré majú za úlohu čítať prekladové pamäte a sprostredkovať ich obsah vo forme kontrolovateľných dvojíc textu.
- Úprava a kontrola – moduly, ktoré majú za úlohu pred-pripraviť, či následne kontrolovať načítané dvojice textu.
- Výstup – moduly, ktoré majú za úlohu poskytnúť užívateľovi výsledky.

6.1 Vstupné dáta

Na načítanie vstupných dát slúžia čítacie moduly, v riešení nazývané Reader moduly. Užívateľ má možnosť si podľa typu a umiestnenia vstupných dát zvoliť (prípadne i sám implementovať) najvhodnejší čítací modul. Samotná realizácia vstupných dát pritom nie je rozhraním obmedzená. Môže to byť súbor na lokálnom disku, ale napríklad i vzdialené úložisko alebo databázový server.[1]

Vzhľadom k tomu, že čítacie moduly sú navonok navzájom kompatibilné, je v jednom projekte možné nielen čítať viacero súborov, ale v prípade potreby i kombinovať rôzne typy vstupných dát a ich umiestnení.

[1] V rámci tejto diplomovej práce sú však implementované iba niektoré vybrané formáty vstupných súborov. Implementácia vzdialených zdrojov či databáz nebola realizovaná.

6.2 Úprava kontrolovaných dvojíc

Na úpravu kontrolovaného textu slúžia modifikačné moduly, v riešení nazývané Filter moduly. Cieľom takejto úpravy je dostať text do formátu, v ktorom samotné hľadanie chýb dokáže pracovať. To zahŕňa napríklad odstránenie formátovacích značiek (HTML alebo RTF značiek), konverziu zástupných symbolov, odstránenie špeciálnych značiek, ale i prípadnú morfológiu a ďalšie podobné úpravy.

6.3 Hľadanie chýb

Na hľadanie chýb slúžia vyhľadávacie moduly, ktoré sú v riešení implementované opäť ako Filter moduly. Z pohľadu logiky aplikácie a rozhrania totiž rozdiel medzi modifikačným filtrom a vyhľadávacím filtrom v zásade nie je. Z pohľadu užívateľa rozdiel závisí iba na určení funkcie samotných modulov. Kombinovanie úpravy a vyhľadávania v jednom module však nie je doporučené, i keď nie je priamo zakázané.[1]

Výhoda takéhoto prístupu spočíva v tom, že rôzne hľadanie chýb môže vyžadovať rôzny stupeň úpravy textu. Užívateľ by mal mať možnosť zostaviť si také poradie modulov, aby potrebná úprava textu (napr. odstránenie formátovania) vždy predchádzala následnému hľadaniu chýb. Zároveň však tento prístup berie do úvahy fakt, že konkrétnu úpravu obvykle využije celá skupina kontrolných modulov. Do tretice je nutné spomenúť, že konkrétny modul môže pre rôzne pravidlá vyžadovať rôzne úpravy textu: Zakiaľ pravidlá hľadajúce napríklad zakázané použitie zástupných symbolov vyžadujú text prakticky neupravený, pravidlá hľadajúce zakázané použitie výrazov vyžadujú text očistený od špeciálnych formátovacích značiek; a jedná sa pritom o rovnaký vyhľadávací modul.

[1] Pokiaľ nejaký modul skutočne potrebuje vstupný text upraviť, mal by to buď urobiť iný, k tomu špeciálne určený modifikačný modul, alebo by mala táto úprava prebehnúť privátne pre daný modul.

6.4 Sprostredkovanie výsledkov

Na sprostredkovanie výsledkov slúžia výstupné moduly, v riešení nazývané Output moduly. Úlohou tohto typu modulov je priebežne zbierať výsledky a poskytnúť ich užívateľovi. Opäť platí, že užívateľ si môže zvoliť i viacero výstupných modulov zároveň, napríklad jeden, ktorý bude vytvárať priebežný log súbor, a druhý, ktorý bude výsledky sprostredkovať interaktívne.

6.5 Vyhľadávanie

Základným prvkom celého riešenia je vyhľadávanie slov a fráz v texte. Či už sa jedná o úpravu vstupného textu modifikačnými filtrami, alebo o samotné vyhľadávanie v rámci vyhľadávacích modulov kontrolujúcich preklad. Moje riešenie na takéto vyhľadávanie používa regulárne výrazy typu PCRE[1].

Výhodou použitia regulárnych výrazov je výrazné zvýšenie možností vyhľadávacích filtrov a zároveň ich zjednodušenie. Napríklad v prípade potreby hľadania celej množiny slov naraz majú napríklad možnosť tieto slová vyhľadávať zároveň, použitím PCRE vlastnosti „alternation“[2].

V prípade, že vyhľadávací filter sprostredkuje možnosť využívať regulárne výrazy priamo užívateľovi, otvára tým možnosť skúsenejším užívateľom definovať zložitejšie a predovšetkým výrazne chytrejšie pravidlá.

6.6 Morfológia

Počas implementácie vyhľadávania som postupne preskúmal viacero možností ako vyhľadávať s ohľadom na morfológiu. Pôvodne preferovaná bola možnosť aplikovať na český text nástroj, tzv. morfologickú analýzu, ktorá by jednotlivé slová previedla na ich základný tvar, prípadne k nim doplnila informácie o pôvodnom tvare.

---

[1] PCRE je skratka pre *Perl Compatible Regular Expressions*. Podrobná dokumentácia výrazov tohto typu je k dispozícií na stránkach jazyka Perl, alebo tiež na stránkach projektu PCRE.

[2] Jedná sa o výraz, ktorý v danom momente hľadá dve a viac alternatív zároveň. Napríklad výraz „jedna|dva“ hľadá zároveň slovo „jedna“ i slovo „dva“, a zastaví sa ihneď, ako narazí na prvý výskyt ktoréhokoľvek z nich.

Zvolený nástroj, tzv. morfo-analysis a morfo-inflector, časť systému MORFO[1], túto úlohu síce zvládol výborne, vyskytli sa však niekoľké netriviálne komplikácie:

- Nekompatibilita v platformách. Systém MORFO bol vyvíjaný pre platformy typu Linux, pre Windows momentálne neexistuje žiaden port. Túto komplikáciu som síce vyriešil použitím subsystému Cygwin, ktorý umožňuje skompilovať a spustiť pod Windows aplikácie vyžadujúce Linuxové prostredie, pre reálne komerčné nasadenie by však toto riešenie mohlo predstavovať výrazne väčší problém.
- Rýchlosť analýzy. Už pri neveľkom testovacom objeme záznamov sa morfologická analýza českého textu prejavila ako neprimerane pomalá.
- Bolo potrebné zasiahnuť priamo do zdrojového kódu systému MORFO, aby jeho výstup napĺňal použiteľnú formu. To ďalej implikuje problémy s licenciou a autorskými právami pri prípadnom nasadení.

Z týchto dôvodov som sa rozhodol preskúmať opačnú možnosť – a síce zaviesť morfológiu priamo do vyhľadávacích pravidiel. Okamžité výhody, ktoré tento prístup priniesol, boli jednorázovosť použitia morfológie (a to v čase tvorby pravidiel); a úplná oddeliteľnosť od zvyšku aplikácie.

Morfológia slova je z pohľadu vyhľadávania v texte totiž iba súborom alternatív k danému slovu. Hľadanie rôznych alternatív pritom nie je až tak výpočtovo náročné a použitím spomenutých regulárnych výrazov dokonca pravdepodobne i omnoho efektívnejšie[2] než kompletná morfologická analýza celej vety.

Ďalšia výhoda plynie i z toho, že morfológia sa aplikuje iba na českú časť záznamu, ktorá však v množstve prípadov nie je v skutočnosti vôbec kontrolovaná.[3] Dá sa preto

[1] Systém MORFO, nástroje pre českú morfológiu, je vyvíjaný na Ústave formální a aplikované lingvistiky, http://ufal.mff.cuni.cz/morfo/.

[2] Napríklad vyhľadávanie všetkých alternatív slova „prohlížeč“ je optimalizovaným regulárnym výrazom možné zapísať ako hľadanie výrazu „prohlížeč(ema?|ích|[uů]m|[eiů]?)“.

[3] Pretože neexistuje pravidlo, ktoré by bolo aplikovateľné na daný záznam. Neaplikovateľnosť sa u drvivej väčšiny pravidiel a záznamov zistí výhradne použitím anglickej časti záznamu a pravidla.

povedať, že presunutím morfológie do pravidiel sa prípadný potrebný výpočet[1] odsúva na najneskorší možný čas, tzv. „lazy vyhodnocovanie“.

Neposlednou výhodou tohto prístupu je spomínaná oddeliteľnosť vyhľadávacej aplikácie od systému poskytujúceho morfológiu. To zo sebou prináša možnosť ponechať aplikáciu na platforme Windows, a zároveň presunúť morfológiu na Linux. Výstup morfológie, už mimo iné prevedený do optimalizovaného regulárneho výrazu, je užívateľovi poskytnutý napríklad cez webové rozhranie, ktoré rozdiely platforiem efektívne odstráni.[2]

Poslednou zaujímavou výhodou tohto prístupu je možnosť aktívne sa do morfológie pri tvorbe pravidiel zapojiť. Napríklad vo výraze „cílová* složka*“ stačí slovo cílová hľadať v ženskom rode[3], čo jednak znižuje výpočtovú náročnosť a zároveň kontroluje prípadné preklepy, ktoré by úplná morfológia mohla prehliadnuť ako správny tvar[4].

6.7 Použité technológie

Jadro aplikácie je implementované v jazyku C, využíva však rozšírenia špecifické pre Microsoft. Aplikácia samotná beží na platformách kompatibilných s MS Windows XP, čo je momentálne obľúbená platforma prekladateľov.[5]

Časť implementácie jadra pochádza z iných projektov. Sú to predovšetkým knižnice na čítanie INI súborov, knižnice na prácu s textom, a knižnice na hľadanie pomocou regulárnych výrazov.

---

[1] Výpočtom sa tu rozumie čas, ktorý buď aplikácia strávi overovaním rôznych alternatív hľadaného slova; alebo čas, ktorý bezpochyby spotrebuje kompletná morfologická analýza celej vety; teda čas, ktorý by ušetrila, keby žiadna morfológia v probléme neexistovala.

[2] Dá sa predpokladať, že profesionálny prekladateľ ľahšie vyrieši problém vzniku linuxového webového serveru, než problémy s platformami a kompiláciou v Cygwin systéme. Riešenie využíva jazyk PHP, ktorý vkladá morfológiu získanú zo systému MORFO priamo do pravidiel. PHP sa pritom dá efektívne využiť prístupom cez webový server, ale i cez príkazový riadok.

[3] Pokiaľ sa nejedná o hľadanie zakázaného výrazu, kde by naopak viac prospel kompletný zoznam všetkých možných tvarov i možných preklepov.

[4] V tomto príklade je podobný preklep dokonca považovaný za notorický – prekladateľ síce opraví slovo „adresář“ na slovo „složka“, neopraví už však prídavné meno pred ním.

[5] Prekladateľské produkty sú obvykle ponúkané práve pre systém MS Windows.

Implementácia modulov, i keď momentálne sú všetky staticky linkované priamo do aplikácie, ponúka možné budúce rozšírenie na podporu dynamicky linkovaných knižníc. Jednotlivé moduly by tak vo výsledku nemuseli zdieľať s jadrom ani rovnaký programovací jazyk.

Implementácia automatizovanej tvorby morfológiou obohatených pravidiel zo slovníka využíva jazyk PHP a systém MORFO. Je kompatibilná s platformami Windows i Linux. V prípade potreby by systém MORFO mal byť zameniteľný i za iný podobný systém.

7 Implementácia modulov

Táto kapitola bližšie popisuje implementované vyhľadávacie moduly. Pre účely tejto kapitoly (a ďalších) budem používať nasledujúce definície, ktoré sa v rovnakom znení objavujú i v samotnej implementácií modulov:

- **záznam** – samostatný záznam z prekladovej pamäte; veta a jej preklad
- **source** – časť záznamu odpovedajúca jazyku originálu; text tejto časti
- **target** – časť záznamu odpovedajúca jazyku prekladu; text tejto časti
- **pattern** – hľadaný textový reťazec alebo PCRE regulárny výraz; slovo „pattern“ je prevzaté priamo z dokumentácie syntaxe jazyka PCRE

7.1 Pravidlá

Moduly, ktoré sú založené na pravidlách, majú niekoľko spoločných vlastností. Obvykle potrebujú pre každé pravidlo najskôr overiť, že je dané pravidlo relevantné k danému záznamu, a následne dané pravidlo aplikovať. Overovanie relevantnosti a aplikovanie pravidla je pritom obvykle otázkou hľadania v zázname, respektíve v jeho jednotlivých častiach. Pravidlo sa preto skladá z dvoch častí:

- **REQ** – časť pravidla zaoberajúca sa overením relevancie pravidla
- **TEST** – časť pravidla zaoberajúca sa aplikovaním pravidla

Každá z týchto častí pravidla je voliteľná. Každá z nich potrebuje ďalej definovať, čo sa má vyhľadať alebo overiť v *target* a *source* texte.[1] Preto sa každá z týchto častí skladá z dvoch voliteľných[2] definícií:

- **TARGET** – definícia patternu, podľa ktorého sa má hľadať v *target* texte
- **SOURCE** – definícia patternu, podľa ktorého sa má hľadať v *source* texte

[1] Vo všeobecnosti by každá časť mala obsahovať informáciu o tom, čo daný modul má v danej chvíli overiť. Momentálne každý modul (založený na pravidlách) v texte vyhľadáva, ale v budúcnosti by touto informáciou mohlo byť napríklad i číslo určujúce počet očakávaných častí vety, počet očakávaných slov, a pod.

[2] Vo všeobecnosti sa nedá povedať, že časť **REQ** sa zaoberá iba *source* textom a časť **TEST** naopak iba *target* textom. Pravidlo napríklad môže poskytovať prevenciu tzv. „false positives“, ktoré sú veľakrát identifikovateľné práve kontrolou druhej časti záznamu.

Každá definícia môže ďalej obsahovať príznaky špecifické pre vyhľadávanie v texte[1], a síce:

- **CASE** – vyhľadávanie sa ma vykonať v móde rozlišujúcom veľké písmená
- **WORD** – vyhľadávanie sa ma vykonať v móde hľadajúcom iba celé slová
- **REGEX** – pattern nie je jednoduchý textový reťazec, ale výraz typu PCRE
- **BASE** – popis patternu, zjednodušená verzia patternu, prípadne zdroj, z ktorého pattern vychádza (napr. pattern pred aplikovaním morfológie)

Súbor pravidiel používa jednoduchú syntax podobnú XML. Príklad pravidla zo súboru pravidiel ilustruje Príklad 1, kde sú použité všetky vyššie uvedené možnosti a príznaky.

```
<rule>
 <req>
  <source>pattern1</source>
  <target case>pattern2</target>
 </req>
 <test>
  <source word regex>pattern3</source>
  <target regex base="description4">pattern4</target>
 </test>
</rule>
```

Príklad 1: Pravidlo zo súboru pravidiel. Ilustruje rozpoznávané možnosti a príznaky.

7.2 Morfológia

V kapitole o architektúre riešenia som priblížil dôvody, prečo sa aplikovanie morfológie na preklad zdá byť menej vhodné, ako aplikovanie morfológie na pravidlá. Presunutie morfológie do pravidiel však so sebou prináša otázku, ako tento fakt zaznamenať v pravidlách.

Práve z toho dôvodu obsahujú definície vyhľadávania atribút **BASE**. Tento môže obsahovať hľadané slovo v základnom tvare, kým samotná definícia obsahuje regulárny výraz zložený zo všetkých alternatív daného slova.[2] Prípadný generátor, či

[1] Opäť, momentálne každý modul (založený na pravidlách) v texte vyhľadáva, ale v budúcnosti by príznaky mohli byť rozšíriteľné o ďalšie položky.

[2] Napríklad `<target regex base="prohlížeč*">prohlížeč(ema?|ích|[uů]m|[eiů]?)</target>`

editor pravidiel tak má možnosť užívateľovi pohodlne ponúknuť možnosť vrátiť sa k pôvodnému slovu pred aplikovaním morfológie.

Príklad pravidla obsahujúceho morfológiu ilustruje Príklad 2.

```
<rule>
 <req>
  <source word regex base="browser*">browsers?</source>
 </req>
 <test>
  <target word regex
    base="prohlížeč*">prohlížeč(ema?|ích|[uů]m|[eiů]?)</target>
 </test>
</rule>
```

Príklad 2: Príklad reálneho pravidla, ktoré požaduje, aby slovo browser bolo preložené ako prohlížeč. Okrem toho, že je morfológií podrobené slovo prohlížeč, je tomu tak i u slova browser.

7.3 Generátor pravidiel MORPHER

Manuálne vytváranie pravidiel má síce výhodu v tom, že pravidlá sú výrazne presnejšie a na mieru použiteľné v danom prostredí, v ktorom boli vytvorené; požadovať však od prekladateľa, aby strávil potrebný čas písaním regulárnych výrazov či filtrovaním výstupu z morfologického slovníka je viac-menej absurdné.

Plne automatizovaný generátor má naopak nevýhodu v tom, že vytvára pravidlá pomerne všeobecné a v nemalej miere i chybové. Jeho hlavnou výhodou je však to, že s jeho pomocou je prekladateľ schopný si pomerne rýchlo a bez veľkej námahy vytvoriť, dopĺňať a následne upravovať slovník nenáročných pravidiel.

Názorným príkladom takého generátoru je PHP generátor **MORPHER**. Jeho hlavnou funkciou je konverzia poskytnutého vstupného anglicko-českého slovníka jednoduchého textového formátu do výstupného súboru pravidiel. Generátor pri konverzií volá systém MORFO[1], a množiny slov prevádza[2] do optimalizovaných[3]

[1] Najprv volá subsystém „morfo-analysis“, ktorý vstupné slová overí a doplní pozičnými tagmi; a následne volá subsystém „morfo-inflector“, ktorý slová doplní o všetky požadované tvary. V tomto procese sú automaticky preskakované slová hovorové a archaické; ale i tvary, ktoré nevyhovujú požadovaným rodom, číslam, prípadne stupňom porovnania.

[2] Napríklad slovo „`aktivní`“ vo výsledku prevedie na výraz „`(ne)?aktivní(m[aiu]?|(ch|ho)?)`“.

[3] Optimalizáciou sa tu myslí predovšetkým skrátenie výrazu z pohľadu vizuálnej dĺžky a zjednodušenie z pohľadu užívateľskej prehľadnosti výsledného výrazu. Samotné regulárne výrazy v princípe podobnú „optimalizáciu“ k zrýchleniu výpočtu nepotrebujú.

regulárnych výrazov. Základné tvary slov vkladá do atribútu **BASE**, aby bolo v prípade existencie editoru pravidiel možné vrátiť sa k nim.

Keďže napojenie na systém MORFO je pri generovaní pravidiel časovo najnáročnejšie, výsledky z tohto systému si generátor ukladá do vlastnej medzi-pamäte uloženej na disku. Pokiaľ prekladateľ postupne upravuje a zdokonaľuje ten istý zdrojový súbor slovníka, generátor volá systém MORFO už iba pre nové výrazy.

7.4 Modul FORCED

Modul s názvom **FORCED** implementuje vyhľadávací filter na presadzovanie konkrétneho prekladu originálu. Filter je použiteľný predovšetkým na kontrolovanie použitia doporučených prekladov slov a fráz, a na kontrolovanie použitia správnej terminológie. Používa k tomu súbor vyššie popísaných pravidiel.

Pre každý vstupný záznam filter v prvej fáze overí, či je **REQ** časť pravidla prítomná v danom zázname. Pokiaľ áno, v druhej fáze overí, či je v zázname prítomná i časť pravidla **TEST**. V prípade neúspechu hlási chybu v zázname.

Príklad pravidla vhodného pre tento modul ilustruje Príklad 2.

7.5 Modul FORBIDDEN

Modul s názvom **FORBIDDEN** implementuje vyhľadávací filter na kontrolovanie zakázaných výrazov. Filter je použiteľný predovšetkým na kontrolovanie zakázaných prekladov a nevhodných fráz. Podobne ako filter **FORCED**, používa k tomu súbor vyššie popísaných pravidiel.[1]

Pre každý vstupný záznam filter v prvej fáze overí, či je **REQ** časť pravidla prítomná v danom zázname. Pokiaľ áno, v druhej fáze overí, či je v zázname prítomná i časť pravidla **TEST**. V prípade úspechu hlási chybu v zázname.

Príklad pravidla vhodného pre tento modul ilustruje Príklad 3.

---

[1] Samotná implementácia filtrov **FORCED** a **FORBIDDEN** je až nápadne podobná. Dôvodom je prakticky identická úloha, kde hlavný rozdiel spočíva v použitých logických operátoroch. Rozdiel v týchto filtroch je predovšetkým v spôsobe použitia a v metodike písania pravidiel.

```
<rule>
 <req>
  <source word regex base="uninstall*">uninstall.*?</source>
 </req>
 <test>
  <target word regex base="deinstal*">deinstal.*?</target>
 </test>
</rule>
```

Príklad 3: Príklad reálneho pravidla, ktoré zakazuje, aby slovo uninstall bolo preložené ako deinstalace. Morfológia do pravidla nie je zanesená, vyhľadávacie výrazy sú však skonštruované tak, aby ju nahradili.

7.6 Modul COUNTER

Modul s názvom **COUNTER** implementuje vyhľadávací filter, ktorý počíta slová v *source* a *target* texte záznamu a porovnáva pomer týchto počtov. Hlavná myšlienka tohto počítania spočíva v tom, že vety, ktoré sú kratšie ako nejaké percento dĺžky originálu zvyknú byť vety neúplné, prípadne hodne voľne preložené, a naopak, vety dlhšie ako nejaké percento originálu zvyknú byť vety zbytočne zložité, prípadne až domýšľajúce si informácie, ktoré v origináli v skutočnosti nie sú.

7.7 Modul SCAN

Modul s názvom **SCAN** implementuje vyhľadávací filter, ktorý overuje, či záznam neobsahuje zvláštne zloženie častí *source* a *target*.

Prvým typickým problémom je záznam, ktorý obsahuje iba jednu z častí. Jedná sa obvykle o chybu, kedy prekladateľ nechtiac zmazal časť záznamu.

Druhým typickým problémom je záznam, kde sú obe časti identické. Jedná sa obvykle o záznamy, ktoré prekladateľ zabudol preložiť.

Pri porovnávaní častí záznamu tento filter ignoruje rozdiely v prázdnych znakoch, rozdiely v interpunkcií a v ďalších drobných odlišnostiach.[1]

[1] V skutočnosti tento modul porovnáva iba znaky, ktoré sú klasifikované ako písmená slov.

8 Výsledky riešenia

Aplikáciu som mal možnosť otestovať na reálnych dátach – starších prekladových pamätiach vytvorených profesionálnym prekladateľom, t.j. pamätiach, od ktorých som očakával nízku mieru chybovosti. Získané výsledky mi pomohli nielen vylepšiť samotnú aplikáciu, ale i ukázať, že aplikácia je schopná priniesť očakávané výsledky.

8.1 Modul FORCED

K tomuto modulu som vytvoril[1] približne 60 pravidiel, ktoré overovali správne použitie terminológie alebo doporučené preklady konkrétnych slov a výrazov. Do pravidiel som zakomponoval rôzne varianty a synonymá prekladu, ako i morfológiu väčšiny použitých slov. Modul som potom overil na testovacích prekladových pamätiach[2] obsahujúcich spolu 1222 záznamov.

Medzi zaujímavé pozitívne výsledky, ktorých bolo celkovo 18, by som zaradil napríklad:

- preklepy v českých slovách
  - „failed to run“ => „se **na**zdařilo spustit“
  - „License Agreement“ => „**Lc**enční smlouva“
- neúplnosť informácie v preklade
  - „You can delete this **file**“ => „Můžete **jej** odstranit“
  - „Component List **displayed at this location**“ => „Seznam součástí“
- zmena štýlu reči až voľný nepresný preklad
  - „**Change** which features are installed“ => „**Možnost výběru** instalovaných součástí“
- nepresný preklad kľúčových slov
  - „Are you sure you want to **cancel**“ => „Opravdu chcete **ukončit**“

[1] Pravidlá som vytváral na základe pravidiel pre nástroj MemoryMining, zaoberajúci sa kontrolovaním prekladových pamätí pomocou „fuzzy“ vyhľadávania. Tieto pravidlá láskavo poskytol pán Štěpán Kvapilík z firmy Hieronymus.

[2] Modul som overil na dvoch profesionálnych prekladových pamätiach, ktoré obe lokalizovali rôzne produkty z anglického originálu do češtiny. Tieto prekladové pamäte taktiež poskytol pán Štěpán Kvapilík z firmy Hieronymus.

Modul do výsledkov priplietol i množstvo falošných poplachov, ktorých bolo celkovo 51. Medzi zaujímavé prípady by som zaradil:

- neúplnosť definície pravidla, chýbajúci možný variant
  - „currently **selected** item“ => „aktuálně **označené** položky“
  - „I **do not** accept the terms“ => „**Ne**souhlasím s podmínkami“
  - „enough **disk** space for“ => „dostatek místa pro“
- voľný, pravdepodobne však korektný preklad
  - „**Select** Yes to...“ => „**Klepnutím na** tlačítko Ano...“
  - „Don't display this **message** again“ => „Tento **dialog** již příště nezobrazovat“
  - „to a temporary **location**“ => „do dočasné **složky**“
- problematické predložky
  - „click the check box **next to** it“ => „zaškrtněte políčko u jejího názvu“

Z ukážok je očividné, že pravidlá by bolo potrebné ďalej upraviť a doladiť, aby sa znížila frekvencia zbytočne triviálnych falošných poplachov. Vo väčšine týchto falošných poplachov bol totiž dôvod poplachu pomerne jednoducho odstrániteľný, a to pridaním chýbajúcich synoným a ďalších akceptovateľných variantov.

Najviac falošných poplachov (24), zaznamenali záznamy obsahujúce názvy, ktoré prekladateľ nepreložil pravdepodobne vedome. Tu je nutné poznamenať, že v 23 prípadoch sa jednalo o záznamy, ktoré obsahovali iba tento názov. Pokiaľ sa takýto záznam do prekladovej pamäte dostane, je diskutabilné, či je skutočne v poriadku takýto názov nepreložiť. Týchto 23 prípadov sa preto dá zároveň považovať i za pozitívne výsledky.

Medzi najviac neefektívne a problematické pravidlá patrili:

- Pravidlo hľadajúce anglické „location“ a požadujúce český preklad „umístění“ alebo „místo“. Pravidlu chýbala možnosť „složka“, ktorá sa vo falošných poplachoch vyskytovala hneď 7 krát a bola vždy správnym prekladom. Po pridaní tejto možnosti by sa pravidlo stalo na testovacích dátach naopak výrazne efektívnym.
- Pravidlo hľadajúce anglické tvary slova „select“ alebo „selection“ a požadujúce český preklad „vybrat“, „vybrání“, „vybraný“, „výběr“, „zvoliť“, „zvolení“ alebo „zvolený“. Pravidlu chýbali možnosti „označit“,

„označení“, „označený“ a „zadat“, ktoré sa vyskytli v 3 prípadoch. Toto pravidlo však zlyhávalo aj na voľných prekladoch typu „Klepnutím na...“, kde náprava nie je zďaleka jednoduchá a počet akceptovateľných variantov by mohol prudko stúpať. Z výsledkov sa dá usúdiť, že problematická časť pravidla bola viac v anglickom „select“, zakiaľ „selection“ by bolo po opravách pravidla už výrazne efektívne.

Zhrnutím, modul našiel 18 pozitívnych výsledkov, 28 falošných poplachov, a 23 prípadov, ktoré sa dajú klasifikovať ako pozitívne i falošné, a to podľa uhlu pohľadu[1]. Efektivita testovaných pravidiel sa teda dá vyjadriť ako 26% až 59% podľa toho, kam diskutabilné prípady zaradíme.

8.2 Modul FORBIDDEN

Podobne ako u predošlého modulu, vytvoril[2] som sadu približne 10 pravidiel, ktoré mali za úlohu overiť zakázanú terminológiu a použitie zakázaných slov. Tu som morfológiu z pravidiel experimentálne vynechal, pretože prípadný preklep v skloňovaní by nemal znamenať, že aplikácia výraz nenájde.

Modul som potom overil na testovacích prekladových pamätiach použitých i u predošlého modulu. Medzi zaujímavé výsledky by som zaradil napríklad:

- použitie zakázaného prekladu
  - „**Uninstalling**“ => „Probíhá **deinstalace**“
  - „**Button**“ => „**Button**“
  - „**Browser**“ => „**Browser**“

[1] Hranica medzi pozitívnymi a falošnými výsledkami sa hľadá pomerne ťažko. Vo všeobecnosti sa dá povedať, že záleží na tom, do akej miery je dovolený voľnejší preklad a čo všetko sa dá považovať za terminológiu. Napríklad, pokiaľ by sme modul použili výhradne na kontrolu terminológie, a terminológia by bola jednoznačne zavedená a prísne vyžadovaná v každej situácií, neefektivita modulu by pravdepodobne vôbec neexistovala, až na triviálne chyby v pravidlách, prípadne morfológií tejto terminológie.

[2] Vytvorenie týchto pravidiel žiaľ predstavovalo problém, pretože poskytnuté pravidlá pre nástroj MemoryMining neobsahovali žiadne pravidlá typu „zakázaný preklad“, nástroj MemoryMining takúto možnosť totiž vôbec neposkytuje. Výsledné pravidlá tak boli v konečnom dôsledku zhrnutím známych nevhodných slov a prekladov (prevzaté slová a chyby ako „meeting“, „click“, „help“), ktoré v prekladovej pamäti v skutočnosti nič nenachádzali.

Nízky počet pravidiel a pravdepodobne i profesionalita testovaného prekladu znamenala, že výsledky tohto modulu neboli nijak početné, modul našiel celkovo 13 výsledkov. Jeho prínos do celkových výsledkov bol znížený navyše i faktom, že 6 týchto chýb našli zároveň i iné vyhľadávacie moduly.

Na druhú stranu, tento modul zaznamenal iba 2 falošné poplachy a stal sa tak nadpriemerne efektívnym. Oba tieto poplachy boli v záznamoch, ktoré obsahovali nepreložené názvy, tu však bolo ich nepreloženie jednoznačne správne.

Problematickým sa tu stalo rozhodnutie vynechať z pravidiel morfológiu a vytvoriť pravidlá tak, aby im na morfológií nezáležalo. I keď tento nápad ako taký bol v skutočnosti v poriadku, jeho implementácia bola nesprávna. Morfológiu bolo totiž vhodné vynechať iba v českej časti pravidla, anglická časť mala morfológiu obsahovať i v tomto prípade.

Medzi pozitívnymi výsledkami, ktorých bolo celkovo 11, sa nachádzalo 5 výsledkov, ktoré boli, podobne ako u predošlého modulu, nepreloženými názvami. Opäť sa jednalo o záznamy, ktoré obsahovali iba tento názov. 4 prípady pritom boli zároveň prípadmi z predošlého modulu.

Zhrnutím, modul našiel 6 pozitívnych výsledkov, 2 falošné poplachy, a 5 prípadov, ktoré sa dajú klasifikovať ako pozitívne i falošné, a to podľa uhlu pohľadu. Efektivita testovaných pravidiel sa teda dá vyjadriť ako 46% až 85% podľa toho, kam diskutabilné prípady zaradíme.

Podľa názoru profesionálneho prekladateľa je práve takýto modul momentálne na trhu nedostatkom, a bol jedným z hlavných očakávaní z jeho strany. Je možné, že po zavedení väčšieho počtu notorických pravidiel bude tento modul viac užitočný.

8.3 Modul COUNTER

Na testovacích dátach, ktoré som mal k dispozícií, som zistil, že rozumná spodná hranica v počte slov pre tieto dáta je 57-65% originálu, s ohľadom na celočíselné

zaokrúhlenie počtu slov smerom nadol. Vety kratšie ako táto hranica boli v 10 prípadoch skutočne prekladmi, ktoré by sa dali považovať za neúplné.[1]

Na druhú stranu, táto hranica obsahovala i 7 falošných poplachov, obvykle situácií, kde prekladateľ použil veľmi voľný preklad; prípadne situácií, kde si anglická gramatika (viac krát) vyžiadala výrazne viac slov na vyjadrenie pomerne jednoduchej veci.[2]

Experimentovanie s touto hranicou nadol či nahor prinášalo vždy viac-menej horšie výsledky, t.j. viac falošných poplachov, alebo menej skutočných upozornení na zvláštny preklad bez zvýšenia efektivity. Bližšie určenie tejto hranice nebolo vzhľadom na nedostatok dát možné, pretože výsledky sa v tomto rozmedzí nemenili.

Na druhej strane škály dĺžok, sa mi rozumnú hornú hranicu nepodarilo nájsť. Hranica menšia ako 200% originálu znamenala vždy výrazné množstvo falošných poplachov – napríklad hranica 190% zaznamenala 26 falošných poplachov oproti 2 pozitívnym a 2 diskutabilným výsledkom.

Naopak hranica 200% už neprinášala žiadne zaujímavé výsledky – 2 falošné poplachy a jeden diskutabilný prípad, kde nebol problémom počet slov, ale voľba významu slov.

8.4 Modul SCAN

Tento modul aktuálne nepoužíva žiadne pravidlá a ani žiadne špecifické nastavenia či konštanty. Výsledky tohto modulu obsahovali 350 nepreložených záznamov.

---

[1] Ako príklad sa dá uviesť preklad „Component List will be displayed here.“ na príliš krátke „Seznam součástí.“. V prípadnej aplikácií, kde bude tento text zobrazený, je pravdepodobne jedno, že český preklad ignoruje časť „bude zobrazený“, každopádne, preklad je to zvláštny.

[2] Ako príklad sa dá uviesť preklad „Are you sure you want to cancel the installation?“ na „Opravdu chcete instalaci zrušit?“. Preklad je samozrejme v poriadku, angličtina si tu však vyžiadala výrazne viac slov na vyjadrenie „opravdu chcete“.

Z týchto 350 záznamov bolo 340 názvov, u ktorých je diskutabilné, či je to skutočne chyba. Zo zvyšných 10 záznamov iba 3 boli evidentné chyby prekladateľa[1], a z toho 1 prípad zároveň odhalili aj moduly **FORCED** a **FORBIDDEN**.

Efektivita tohto modulu sa teda dá vyjadriť ako 1% až 98% podľa toho, kam veľké množstvo diskutabilných prípadov nepreložených názvov zaradíme. Vzhľadom k tomu, že prekladanie názvov býva u podobných produktov[2] obvyklé, je možné uvažovať o efektivite 98%.

8.5 Zhrnutie výsledkov všetkých modulov

Aplikácia našla nemalé množstvo záznamov (36 prípadov), ktoré vyžadujú ďalšiu pozornosť prekladateľa. V ďalšej časti nájdených prípadov (340 záznamov) sa jedná o diskutabilné nepreložené názvy. V časti prípadov sa jedná o falošné poplachy (37 prípadov).

Falošné poplachy boli spôsobené predovšetkým neúplnými pravidlami, alebo príliš jednoduchým návrhom niektorých metód hľadania chýb. Kým náprava niektorých pravidiel bola očividná a jednoduchá; nápravu problematických metód by predstavovala pravdepodobne iba úplná zmena prístupu, či rozšírenie komplexnosti danej metódy.

8.6 Reálny slovník a generátor pravidiel MORPHER

Generátorom **MORPHER** som zo slovníka[3] vygeneroval sadu približne 1000 pravidiel, a tieto pravidlá som použil pomocou modulu **FORCED** na kontrolu prekladovej pamäte

[1] V jednom zaujímavom prípade tento modul odhalil záznam, v ktorom *target* i *source* časti obsahovali rovnakú českú vetu. Jednalo sa očividne o chybu prekladateľa, ktorý po preložení nedopatrením prepísal anglickú vetu jej českým ekvivalentom.

[2] Rozumie sa „produktov podobných tomu, ktorý testovacia prekladová pamäť prekladala“.

[3] Slovník česko-anglických pravidiel, pôvodne určený pre nástroj MemoryMining, poskytol pán Štěpán Kvapilík z firmy Hieronymus. Tento slovník mal predstavovať zbierku požadovaných prekladov anglických slov do češtiny.

o 867 záznamoch[1]. Nezávisle na pravidlách som požiadal prekladateľa o manuálnu kontrolu tej istej prekladovej pamäte[2].

Výsledky automatickej kontroly som následne porovnal s výsledkami manuálnej kontroly. Prekladateľ počas manuálnej kontroly našiel 138 chýb, z ktorých 107 našiel zároveň i modul **FORCED** v kombinácií s modulmi **COUNTER** a **SCAN**. Podiel na nájdených chybách mali moduly **COUNTER** a **SCAN** však iba minimálny, a to 7, resp. 4 chyby. Úspešnosť kontroly s pohľadu prekladateľa sa teda dá vyjadriť ako 78%.

Medzi hlavné typy chýb, kde modul **FORCED** s použitím automaticky generovaných pravidiel uspel, patrilo nedodržanie terminológie a preklepy v slovách. Naopak medzi hlavné typy chýb, kde modul **FORCED** zlyhal, patrilo pridávanie konkretizujúcich slov do českého prekladu[3], a to v 9 prípadoch; v 8 prípadoch sa jednalo o terminológiu, ktorá nebola zahrnutá v slovníku; v 7 prípadoch sa jednalo o zmenu osoby alebo o nevhodnú zámenu slovesa za podstatné meno[4]; v 3 prípadoch sa jednalo o príliš voľný preklad; v 2 prípadoch o nevhodnú zmenu trpného tvaru na činný; a zvyšné 2 prípady obsahovali gramatické chyby, ktoré modul nemal možnosť odhaliť.

Po porovnaní výsledkov mal prekladateľ možnosť preskúmať, aké ďalšie chyby našla automatická kontrola. Väčšina týchto chýb boli falošné poplachy, avšak v 2 prípadoch sa automatickej kontrole poradilo odhaliť preklepy, ktoré prekladateľ pri manuálnej kontrole prehliadol; v 5 prípadoch automatická kontrola odhalila prehliadnuté

---

[1] Profesionálna prekladová pamäť, ktorá lokalizovala nemenovaný produkt z anglického originálu do češtiny. Prekladovú pamäť taktiež poskytol pán Štěpán Kvapilík z firmy Hieronymus.

[2] Keďže sa malo jednať predovšetkým o kontrolu používania terminológie, prekladateľ sa snažil pri kontrole prísne dodržovať dohodnutú terminológiu.

[3] Napríklad pridanie slova „súbor“ i napriek tomu, že anglická veta súbor nijak nespomína a dovoľuje tým jej využitie i v prípade adresáru. V niekoľkých prípadoch takto prekladateľ priamo zúžil význam pôvodnej vety, čo by sa pri rozšírení produktu mohlo nevyplatiť.

[4] Predovšetkým sa jednalo o zámenu slovesa za podstatné meno v záznamoch, ktoré boli pravdepodobne kontextovými nápoveďami pre položky v hlavnom menu, napríklad „Create new folder“ a „Vytvoření nové složky“. I keď by sa na prvý pohľad mohlo zdať, že preklad je v poriadku, menu v skutočnosti obsahuje „akcie“, a je preto vhodnejšie používať v názvoch a nápovediach slovesá miesto podstatných mien, ako je tomu v anglickej časti záznamu.

nedodržanie terminológie; a v ďalších niekoľkých prípadoch prekladateľ zauvažoval, že by možno bolo vhodné i tieto záznamy poopraviť[1].

8.7 Zhrnutie efektivity

Zhrnutím výsledkov spomeniem, že kým úspešnosť v hľadaní chýb oproti prekladateľovi bola 78%, počet falošných poplachov voči skutočným chybám bol približne 50%, a automatická kontrola dokázala nájsť ďalších 5% chýb, ktoré by inak prekladateľ prehliadol. Z výsledkov je možné zhodnotiť, že nástroj skutočne dokáže prekladateľovi pomôcť pri kontrole prekladu, i keď ho nedokáže úplne nahradiť.

8.8 Názor prekladateľa na riešenie

Riešenie, spolu s modulmi a pravidlami som zaslal prekladateľovi s prosbou o vyjadrenie sa k jeho zaujímavosti, použiteľnosti a prípadne i k tomu, v akej miere je moje riešenie nahraditeľné inými, dnes dostupnými riešeniami.

Vo svojom vyjadrení konštatuje, že riešenie zhruba napĺňa jeho predstavu o nástroji, ktorý na trhu momentálne stále chýba. Pozitívne zhodnotil, akým spôsobom je do pravidiel vložená morfológia; a taktiež, že nástroj beží neinteraktívne a až následne zobrazí výsledky[2].

Menšiu výčitku mal voči grafickému usporiadaniu výstupu, ktoré by si pre praktické účely predstavoval trochu inak.

[1] Tu však už bol problém nájsť hranicu toho, čo je ešte vhodné považovať za chyby, a čo je len iná, ale správna formulácia originálnej vety.

[2] Pôvodná obava, že kontrola bude príliš pomalá na to, aby bolo vhodné interaktívne zobrazovať výsledky, sa nenaplnila. Generovanie približne 1000 pravidiel síce trvalo asi 20 minút na bežnom osobnom počítači, avšak samotná kontrola už potom trvala rádovo iba desiatky sekúnd.

9 Problémy riešenia a možnosti ďalšieho vývoja

Medzi hlavné problémy, na ktoré som počas implementácie narazil patrí:

9.1 Neohybná logika pravidiel

Počas tvorby pravidiel som zaznamenal, že jednoduchá logika vo forme „ak A a B, potom C a D“ nie je pri kontrole prekladu vždy postačujúca. Niekoľko krát som narazil na problém so zbytočnými falošnými poplachmi, a výnimkami a nejednoznačnosťou medzi češtinou a angličtinou.[1] Pritom riešenie týchto problémov by sa mnohokrát dalo zapísať ako skupina rôznych vyhľadávaní v texte, pozitívnych či negatívnych, a to ešte pri overovaní aplikovateľnosti pravidla.

Ako vhodný začiatok riešenia tohto problému by som navrhol zmenu formátu pravidiel tak, aby každé pravidlo malo možnosť vyhľadávať existenciu či neexistenciu ľubovoľného množstva slov a výrazov, jak v anglickej, tak v českej časti záznamu. Tým by sa mimo iné zmazal i rozdiel medzi modulmi overujúcimi správne použitie terminológie a modulmi vyhľadávajúcimi použitie zakázaných prekladov – záležalo by iba na definícií pravidla.

9.2 Generalizácia a unifikácia medzi-modulovej komunikácie

Generalizované rozhranie čítania vstupu dát produkuje unifikované záznamy, ktoré nie je možné ďalej vnímať ako záznamy pochádzajúce z konkrétneho vstupného formátu. Napríklad i preto, že v budúcnosti sa môžu objaviť úplne nové vstupné formáty. Výstupné moduly sú takto nútené predkladané záznamy chápať ako záznamy z ľubovoľného zdroja.

To bolo cieľom návrhu. Má to však nepríjemnú vedľajšiu vlastnosť. Výstupné moduly nie sú takto schopné pracovať so vstupným súborom (napríklad otvoriť problémový súbor v správnom programe na správnom zázname). Správnou otázkou je, ako táto vedľajšia vlastnosť prekáža v prípadnom nasadení. Pokiaľ aplikácia nájde a navrhne

[1] Napríklad „Setup wizard“ je „průvodce instalací“, ale samotné „setup“ znamená „nastavení“.

chybu, a prekladateľ túto chybu bude chcieť opraviť, mal by mať možnosť sa jednoducho „preniesť“ do programu, kde tak môže vykonať.

Na druhú stranu, momentálne je aplikácia schopná čítať iba formáty, ktoré sú v skutočnosti exportmi z programov, ktoré prekladateľ používa. Otvorenie a prípadná modifikácia tohto exportu by pre prekladateľa i tak znamenala, že musí tento export spätne importovať do pôvodného programu. Momentálne sa preto zdá, že toto úskalie nie je v skutočnosti problém unifikácie čítaných záznamov, ale problém toho, že aplikácia nečíta natívne formáty profesionálnych produktov.[1]

Ako riešenie tohto problému, v prípade, že by existovala požiadavka na možnosť otvorenia prípadného záznamu priamo v nejakom programe, by som navrhol, aby každý vstupný modul poskytoval špeciálnu spätnú funkciu, tzv. callback, schopnú daný záznam v prípade potreby „otvoriť“. Výstup by tak dostal príležitosť ponúknuť túto možnosť užívateľovi. Čo by táto akcia pre užívateľa znamenala by však záležalo na vstupnom module.

Druhou možnosťou, ako tento problém riešiť je zavedenie novej skupiny modulov, ktoré by poskytovali možnosť otvárať záznamy. Vstupný modul by zanechal jednoznačnú a dostatočnú informáciu o tom, o aký záznam sa jedná, aplikácia by podľa toho zvolila správny otvárací modul, a použila ho na otvorenie daného záznamu.

Výhoda tohto riešenia oproti predošlému spočíva predovšetkým v tom, že môže existovať viacej rozličných spôsobov, ako záznam konkrétneho formátu otvoriť.[2] Užívateľ dostane možnosť si vybrať požadovaný spôsob podľa toho, aké moduly aplikácii poskytol v nastavení aktuálneho projektu.

---

[1] Dokumentáciu natívnych binárnych formátov profesionálnych produktov sa mi vôbec nepodarilo nájsť. Dokumentáciu exportných formátov som mimo iné tiež nenašiel, našťastie sú to formáty jednoduché a textové.

[2] Napríklad otvorenie v programe, ktorý daný formát používa; alebo otvorenie v obyčajnom textovom editore. Prípadne i otvorenie v nástroji, ktorý so záznamom vykoná nejakú úlohu.

9.3 Perzistentnosť úprav v texte záznamov

Filtre sa v delia na dva druhy, modifikačné a vyhľadávacie. Modifikačné filtre majú za úlohu upraviť v danej chvíli záznam podľa svojej špecifikácie. Tento upravený záznam je potom poskytovaný ďalším filtrom v poradí. To môže viesť ku konfliktu záujmov vo filtroch vyhľadávacích – kým jeden filter potrebuje k svojej funkčnosti úpravu A, druhý filter potrebuje úpravu B, pričom úpravy A a B nie sú navzájom kompatibilné.

Momentálne pravdepodobne neexistuje takáto kombinácia vyhľadávacích filtrov, existuje však takáto kombinácia vstupných formátov.

Zakiaľ formát systému Trados zdá sa preferuje na formátovanie textu použitie RTF značiek, systém TranslationManager preferuje použitie HTML značiek. Aplikácia síce poskytuje filtre na odstránenie RTF značiek rovnako ako na odstránenie HTML značiek, pokiaľ však užívateľ zvolí oba tieto vstupné formáty v jednom projekte, nastane komplikovaná situácia. Na niektoré záznamy potrebuje užívateľ aplikovať RTF filter, a na iné záznamy zase HTML filter. Aplikovanie oboch filtrov za sebou je síce možné, a pre väčšinu záznamov dokonca i akceptovateľné, neplatí to však vo všeobecnosti. Trados totiž špeciálne HTML znaky nijak nekóduje, a prípadný HTML filter tieto znaky interpretuje v tej chvíli nesprávne, obvykle odstránením časti textu.

Dá sa našťastie predpokladať, že užívateľ nebude mať potrebu kombinovať v jednom projekte rôzne vstupné formáty. Nebude mať preto ani potrebu kombinovať rôzne čistiace filtre.

Pokiaľ by v budúcnosti vznikol konflikt priamo medzi vyhľadávacími filtrami, riešením by mohlo byť zavedenie skupín do filtrov. Skupina by sa navonok javila ako nemodifikačný filter, t.j. akékoľvek modifikácie v rámci skupiny by boli privátne pre danú skupinu. Zvyšok filtrov mimo skupiny by mohol naďalej používať pôvodné znenie záznamu.

9.4 Pokročilosť zápisu pravidiel

Vzhľadom k tomu, že pravidlá, ktoré sú schopné efektívne priniesť výsledky sú obvykle pravidlá obsahujúce zložitejšie regulárne výrazy, novou požiadavkou na tvorbu týchto pravidiel sa stáva znalosť syntaxe regulárnych výrazov. I keď nástroj na optimalizáciu regulárnych výrazov dokáže človeku výrazne pomôcť, hlavná zodpovednosť za korektnosť výrazov stále zostáva v rukách tvorcu pravidla. Pokiaľ tvorca pravidiel nemá s regulárnymi výrazmi dostatočné skúsenosti, nie je schopný zaručiť ich správnosť a odladiť prípadné problémy.

Čiastočným riešením by mohol byť napríklad pokročilý vizuálny nástroj, ktorý by regulárny výraz zobrazil farebne, napovedajúc tak užívateľovi, ako je daný výraz chápaný z pohľadu syntaxe regulárnych výrazov.[1]

Iným riešením by mohlo byť vytvorenie vizuálneho užívateľského nástroja, ktorý by korektné regulárne výrazy dokázal tvoriť úplne sám, napríklad na základe alternatív získaných z morfológie.[2] Takýto nástroj by sa eventuálne mohol stať i kompletným riešením editácie súborov pravidiel, a prípadný tvorca pravidiel by následne nemusel ovládať ani syntax súboru pravidiel, ani syntax regulárnych výrazov. V momentálnej fáze vývoja aplikácie, ale i v kontexte tejto práce, sú však takéto pomôcky málo prioritné.[3]

9.5 Rýchlosť výpočtu a paralelizácia výpočtu

Počas implementácie som si uvedomil, že celý výpočet je navrhnutý sekvenčne. To v dnešnej dobe znamená, že aplikácia nedokáže rozumne využiť viac procesorov, čo už dnes začína byť dostupný štandard. Výpočet beží v jedinom vlákne, ktoré najskôr načíta jeden vstupný záznam, potom postupne zavolá všetky filtre, a v prípade chýb absolvuje výstup.

---

[1] Také nástroje dnes už existujú a dajú sa kúpiť.

[2] Takýto nástroj je v neinteraktívnej forme súčasťou tejto práce.

[3] Jedná sa totiž o externé nástroje, ktorých jedinou a hlavnou úlohou je príjemné grafické rozhranie k už existujúcej funkcionalite. Možnosti, ktoré aplikácia ponúka nijak nemenia.

Ako riešenie by sa dalo navrhnúť označenie filtrov, ktoré sú voči záznamom konštantné, t.j. nijak ich nemodifikujú, a tieto filtre potom vykonať paralelne. Inou možnosťou by bolo paralelizovať výpočet nad jednotlivými záznamami. I keď samotné čítanie záznamov nemusí byť paralelizovateľné, spustenie filtrov nad každým načítaným záznamom by paralelné byť mohlo. Tu potom vzniká už len otázka synchronizácie poradia výstupu, ktorý by mal v zásade kvôli užívateľovi dodržať poradie na vstupe.

10 Záver

Vo svojej práci som preskúmal, aké sú najčastejšie a najpálčivejšie problémy pri profesionálnom preklade z anglického do českého jazyka. Špeciálne som sa zameral na oblasť týkajúcu sa dnes veľmi obľúbených prekladových pamätí, a ich využitie pri preklade verziovaných dokumentov, ako napríklad lokalizácií produktov a ich dokumentácie.

Analyzoval som typy chýb, ktoré prekladatelia majú tendenciu robiť, a pokúsil som sa zhrnúť, ako si dnešní prekladatelia predstavujú nástroj, ktorý by im v ich práci pomohol tieto chyby efektívne nájsť.

Navrhol som aplikáciu, ktorá tieto chyby automaticky vyhľadáva na základe vopred skonštruovaných pravidiel. Na reálnych testovacích dátach som následne overil, že aplikácia už i s malým počtom pravidiel dokáže efektívne nájsť chyby v prekladovej pamäti, ktorá bola produktom profesionálneho prekladateľa.

Ako najzaujímavejšia a zároveň najproblematickejšia sa spočiatku zdala byť morfológia. Riešenie ale ukázalo, že i česká morfológia sa dá, za použitia elektronického morfologického slovníku, jednoducho, jednorazovo a efektívne zakomponovať priamo do vyhľadávacích pravidiel; samozrejme za cenu zvýšenia zložitosti zápisu týchto pravidiel, a teda nutnosti vyšších znalostí pri ich tvorbe.

Počas riešenia som narazil na drobné technické nedostatky v pôvodnom návrhu architektúry, ktoré by v prípade potreby vyžadovali mierne pozmenenie návrhu. Jednalo sa však predovšetkým o nedostatky v užívateľskom rozhraní, založené na príliš veľkej všeobecnosti návrhu architektúry, nie o nedostatky v samotnom hľadaní chýb v preklade.

Pokračovať by sa ďalej dalo skúmaním ďalších pokročilejších možností kontrol, napríklad kontrolou slovosledu vo vetách alebo experimentovaním so štýlom reči.

Použitá literatúra

Abekawa Takeshi and Kageura Kyo What Prompts Translators to Modify Draft Translations? An Analysis of Basic Modification Patterns for Use in the Automatic Notification of Awkwardly Translated Text [Conference] // IJCNLP 2008: Third International Joint Conference on Natural Language Processing. - Hyderabad, India : [s.n.], 2008. - pp. 241-248.

Cruz-Lara Samuel [et al.] Interoperability between translation memories and localization tools by using the MultiLingual Information Framework [Conference] // EAMT-2006: 11th Annual Conference of the European Association for Machine Translation. - Oslo, Norway : [s.n.], 2006. - pp. 132-139.

Flournoy Raymond and Duran Christine Machine Translation and Document Localization at Adobe: From Pilot to Production [Conference] // MT Summit XII: proceedings of the twelfth Machine Translation Summit. - Ottawa, Ontario, Canada : [s.n.], 2009. - pp. 425-428.

Hodász Gábor and Pohl Gábor MetaMorpho TM: a linguistically enriched translation memory [Conference] // International workshop: Modern approaches in translation technologies. - Borovets, Bulgaria : [s.n.], 2005. - pp. 26-30.

Hua Wu [et al.] Improving Translation Memory with Word Alignment Information [Conference] // MT Summit X: Conference Proceedings. - Phuket, Thailand : [s.n.], 2005. - pp. 364-371.

Jutras Jean-Marc An Automatic Reviser: The TransCheck System [Conference] // ANLP-NAACL-2000: proceedings of the Sixth conference on Applied Natural Language Processing and 1st Meeting of the North American Chapter of the Association for Computational Linguistics. - Seattle, Washington : [s.n.], 2000. - pp. 127-134.

Lagoudaki Elina The Value of Machine Translation for the Professional Translator [Conference] // Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas. - Waikiki, Hawai'i : [s.n.], 2008. - pp. 262-269.

Macklovitch Elliott TransCheck - or the Automatic Validation of Human Translations [Conference] // MT Summit V. - Luxembourg : [s.n.], 1995.

Richardson Stephen Microsoft Machine Translation: From Research to Real User [Conference] // MT Summit XI. - Copenhagen, Denmark : [s.n.], 2007. - p. 33.

Smith Ross Your own memory? [Article] // The Linguist. - February-March 2008. - 47. - pp. 22-23.